

P21738.P03

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :N. KIDO et al.

Appl No. : Not Yet Assigned

PCT Branch

I.A. Filed : March 29, 2001

PCT/JP01/02637

For : MATCHED FILTER AND CORRELATION DETECTION METHOD

CLAIM OF PRIORITY

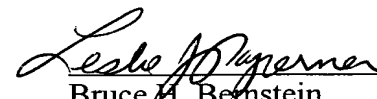
Commissioner of Patents and Trademarks

Washington, D.C. 20231

Sir:

Applicant hereby claims the right of priority granted pursuant to 35 U.S.C. 119 based upon Japanese Application No.2000-099306, filed March 31, 2000. The International Bureau already should have sent a certified copy of the Japanese application to the United States designated office. If the certified copy has not arrived, please contact the undersigned.

Respectfully submitted,
N. KIDO


Bruce A. Bernstein
Reg. No. 29,027

Reg No
33,329

November 27, 2001
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191

THIS PAGE BLANK (USPTO)

明 細 書

(発明の名称)

マッチドフィルタおよび相関検出方法

5

(技術分野)

本発明は、スペクトル拡散通信方式の同期確立処理において使用されるマッチドフィルタと、マッチドフィルタを用いた相関検出方法に関する。

10 (背景技術)

近年、移動体通信において、周波数利用効率が高く、高速かつ高品質なデータ通信が可能となるスペクトル拡散通信、特にCDMA（符号分割多元接続）方式が主流になっている。

15 一つの基地局がカバーするエリアをセル（cell）という。セルの半径を大きくできれば、基地局の数を削減できる。よって、セル半径は、移動通信システムの中では重要な項目であり、実際には、セル半径は、数十kmに及ぶ場合がある。

セル半径が数十kmにもなると、移動体が基地局の近くに位置する場合と、セルの境界付近に位置する場合とでは、移動体から基地局までの距離が大きく異なる。したがって、基地局が受信する信号の遅延量は、大幅にばらつく。

20 CDMA方式の移動体通信において、移動端末機と基地局間で通信を開始する場合には、基地局は、まず、十分な大きさのセル半径を考慮して、高速な初期同期処理を行う必要がある。

例えば、移動端末機は通信のターゲットである基地局に対し通信許可を求め
25 る際、基地局は、移動端末機がどのくらいの距離、離れた位置にあるかを、基準信号に対する伝搬遅延時間を測定することにより知る。

すなわち、基地局から所定のタイミング（基準タイミング）で移動局に対して基準信号を送信し、それに対応して移動局側からACK（応答信号）が戻ってくるまでの時間（これが伝播遅延時間である）を測定することにより、移動局までのだいたいの距離がわかる。この結果、基地局は、基地局が発した信号
5 に対する移動局からの応答信号を検出する際の、相関検出におけるサーチ範囲を、ある程度、絞り込むことができる。

伝搬遅延時間は、マッチドフィルタ回路で算出した遅延プロファイル情報を参照して求める。

マッチドフィルタで遅延プロファイルを生成する場合、いわゆるデータスキ
10 ャン・コード固定の逆拡散演算を行う。すなわち、マッチドフィルタに、対象シンボルの逆拡散演算に必要な拡散コードをセットし（コード固定）、伝搬遅延時間を考慮して決定されるサーチ区間における受信データを連続的に入力して相関値を算出する。

図21は、従来のマッチドフィルタの構成を示す。

15 コード発生器106により生成された拡散コードは、コードレジスタ107に格納される。コードレジスタ107は、256個のシフトレジスタ104bからなる。

次に、受信データが、シフトレジスタ105に、シリアルに入力される。受信データが1チップ入力される毎に、逆拡散演算部109は、シフトレジスタ
20 105の平行出力と、コードレジスタ107の出力とを乗算する。逆拡散演算部109は、256個の乗算回路108を有する。

そして、積分演算部103（積分回路110を有する）は、逆拡散演算部109の平行出力を積分し、1シンボル分のデータの相関値を算出する。

サーチ範囲（遅延時間の幅）が0チップ～255チップ（1シンボル相当）
25 であるとする、データを1チップシフトさせる毎に上述の演算を行うのであるから、合計で256回の演算が必要である。

マッチドフィルタは、1シンボルを単位として相関を検出するため、1つのマッチドフィルタでは、最大で、1シンボル分の伝播遅延をもつ受信信号しか処理できない。

- ここで、セル半径が大きいために、サーチ範囲（遅延時間の幅）が、0～5
 5 1チップ（2シンボル相当）まで増大したとすると、1つのマッチドフィルタでは対応できず、図22Aに示すように、2つのマッチドフィルタ904、906を用いることになる。

図22Aの2つのマッチドフィルタ904、906は、それぞれ、異なる基準タイミングで動作し、1シンボル相当の遅延プロファイルを出力する。

- 10 図22Bは、2つのマッチドフィルタの動作を説明するためのタイミングチャートである。

図示されるように、第1の基準タイミングでマッチドフィルタ904を動作させ、第1の基準タイミングから1シンボル相当の時間だけ遅れた第2の基準タイミングでマッチドフィルタ906が動作する。

- 15 第1シンボルの処理を例にして、各マッチドフィルタの動作を説明する。

時刻T1において、マッチドフィルタ904に第1シンボル用の拡散コードをセットし、1チップずつ入力データをずらしながら、第1シンボルについての相関演算を行う。0チップ～255チップまでのサーチ（前半のサーチ）は、時刻T2に終了する。

- 20 続いて、時刻T2（第2の基準タイミング）において、マッチドフィルタ906に第1シンボル用の拡散コードをセットし、1チップずつ入力データをずらしながら、時刻T3まで、第1シンボルについての相関演算を行う。256チップ～511チップまでのサーチ（後半のサーチ）は、時刻T3に終了する。

- 25 なお、時刻T2に、マッチドフィルタ904には、第2シンボル検出用のコードがセットされ、第2シンボルについてのサーチが開始される。

このようにして求められた、第1シンボルについての0～255チップまで

のサーチ結果と、256チップ～511チップまでのサーチ結果を1つにまとめることにより、第1シンボルについての、2シンボルに相当する時間の遅延プロファイルが得られる。

- しかし、複数のマッチドフィルタを用いると、回路規模が大きくなり、また、消費電力も極端に増大する。すなわち、LSIチップの占有面積が増大すると共に、コストアップや消費電流の増大を招く。

- しかし、現実には、基地局のセル半径（カバー範囲）を大きくしなければならぬ傾向にあり、これにより、伝播遅延も大きくなり、それらの伝播遅延を全てカバーするためには、マッチドフィルタの数も、さらに増やさなければならぬ。

- また、現状では、移動端末機および基地局において、さまざまな条件における遅延プロファイル生成演算を行いたい場合にも、複数のマッチドフィルタを用いて対応している。この場合にも、回路規模の増大、コストアップおよび消費電力の増大が問題となる。

- このように、従来のマッチドフィルタを用いた相関検出は、マッチドフィルタの処理能力が低く、現状の要求を満足させることが困難である。

本発明の目的は、マッチドフィルタを用いた相関検出処理の効率（処理スピード）を格段に向上させ、これにより、回路規模の増大や消費電力の問題を解消することである。

- すなわち、本発明を用いると、基地局のセル半径が大きい場合や、移動局において数多くの条件の遅延プロファイル演算を行いたいような場合においても、基本的に、1つのマッチドフィルタで対応できるようになる。

（発明の開示）

- 本発明では、マッチドフィルタにおいて、コードを固定し、データをスキャンするタイプの逆拡散演算を行わない。その代わりに、データを固定し、コー

ドをスキャンするタイプの逆拡散演算を行う。

データを固定し、コードをスキャンする逆拡散演算とは、所定期間だけ受信データを固定し、その固定されたデータに対して、伝搬遅延に応じた拡散コードを次々と乗算して、相関値を求める方式である。

- 5 従来、マッチドフィルタによる相関演算の処理能力が低いのは、受信データを入力するスピード（すなわち、シフトレジスタの転送スピード）が遅く、この遅いスピードによって、相関検出処理の全体のスピードが制限されてしまうからである。つまり、マッチドフィルタのハードウェアがもつ処理能力を生かしきれていないのが原因である。

- 10 そこで、本発明では、受信データを固定し、拡散コードをスキャンする方式を採用する。これにより、マッチドフィルタにデータを入力することに起因するスピード制限は無くなる。

- 一方、拡散コードの生成や、乗算・加算といった算術演算は、シフトレジスタのデータ転送とは比べものにならないくらい、速く行うことができる。したがって、ハードウェアのもつ処理能力が、相関検出処理の能力を決めることになる。したがって、ハードウェアの潜在能力を最大限に発揮させることができ、これにより、従来にない、効率的な相関演算処理が可能となる。

- したがって、データスキャン・コード固定方式の逆拡散演算による遅延プロファイル作成を複数のマッチドフィルタを用いて行った場合と全く同じ結果が、
20 1つのマッチドフィルタを用いて実現できる。例えば、基地局の初期同期確立のために使用した場合、移動局からの伝播遅延が1シンボルに相当より時間よりも長い場合であっても、マッチドフィルタ1個で対応することが可能となる。

- ただし、本発明のように、入力データを固定した場合、1シンボル長のデータ（256チップのデータ）が、すべて一つのシンボルに属するデータで構成
25 されているとは限らない。つまり、一つ前のシンボルのデータビットと、現在のシンボルのデータビットとが混在している可能性が高い。相関値は、一つの

シンボルについての逆拡散の結果を積分して得られるのであり、異なるシンボルについての部分的な逆拡散結果同士を積分しても、相関値は求めることができない。よって、一つ前のシンボルについての積分演算と、現在のシンボルについての積分演算は、明確に区別して別個に行う必要がある。

- 5 したがって、本発明では、積分演算区間を、シンボルの区切りに対応させて常に変更していく必要がある。つまり、1回の逆拡散処理により得られるデータを、一つ前のシンボルと現在のシンボルとの境界の位置に対応させて2つに分割し、一つ前のシンボルについての逆拡散の結果同士を積分し、また、現在のシンボルについての逆拡散の結果同士を積分する。具体的には、逆拡散コードを1チップだけシフトさせる毎に、積分範囲の終端（および始端）も1チップだけシフトさせ、その積分範囲を随時、変更する。
- 10

各積分により得られるデータは、あくまで部分的な積分結果である。現在のシンボルについての積分結果（部分的な積分結果）は、メモリに一時的に保存する。一方、一つ前のシンボルについての積分結果（部分的な積分結果）のデータは、前回の相関演算処理の際に、メモリに保存されている部分的な積分結果のデータと合算され、これにより、一つ前のシンボルについての相関値が求められる。

15

また、メモリに保存された、現在のシンボルについての部分的な積分結果のデータは、次の相関検出処理において、算出される部分的な積分結果のデータと合算され、そのときに、相関値が求められることになる。

20

本発明のマッチドフィルタにおける演算方法の一つの好ましい態様では、入力データを一時的保持回路に保持してデータを固定すると共に拡散コードをスキャンする方式で逆拡散を行う。そして、得られた逆拡散結果のデータを、拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前にある部分（前半のシンボル部分）と後にある部分（後半のシンボル部分）とに区別し、各部分について逆拡散結果を積分する。そして、前半のシンボル部

25

分に関しては、先に実施された同様の処理の結果として保持されている、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果と加算する。一方、後半のシンボル部分の積分結果は一時的に蓄積しておき、次に同様の処理を実施した結果として得られる、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果に加算し、これによって、シンボル単位の相関を検出する。そして、検出されたシンボル単位の相関値を比較し、最大の相関を示すものを検出することにより、入力データの実際の遅延量が検出される。

また、本発明のマッチドフィルタの一つの好ましい態様では、入力データを蓄積する一時的保持回路と、1チップずつ拡散コードをスキャンしながら一時的保持回路に保持されている入力データに乘算する逆拡散演算回路と、拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前にある部分（前半のシンボル部分）と後にある部分（後半のシンボル部分）の各々の部分について、逆拡散結果を積分する積分回路と、後半のシンボル部分の積分結果を一時的に蓄積する蓄積回路と、前半のシンボル部分に関する積分結果を、先に実施された同様の処理の結果として蓄積回路に保持されている、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果に加算し、1つのシンボルについての相関値を出力する加算回路と、を設ける。

本発明のマッチドフィルタは、基地局および移動局において受信されたスペクトラム拡散変調信号の相関検出用として使用できる。本発明によれば、基地局システムにおける対応セル半径の拡大や、様々な条件の遅延プロファイル演算処理を、従来より格段にスケールダウンされた回路でもって、容易に実現できる。そして、本発明のマッチドフィルタの演算回路を内蔵した相関検出回路を、基地局や移動端末機などの移動体通信システムに搭載することにより、演算方法の改善と演算スピードの高速化の結果として、低コストで低消費電力の移動体通信システムが実現される。

(図面の簡単な説明)

図 1 A は、従来の、コード固定・データスキャン方式の相関検出方法を説明する
5 ための図であり、

図 1 B は、本発明のマッチドフィルタにおける演算（コードスキャン・データ
固定方式）の特徴を説明するための図であり、

図 2 は、本発明のマッチドフィルタにおける演算（コードスキャン・データ固
定方式）における、受信データの遅延量と、拡散コードおよび積分区間との関
10 係を具体的に説明するための図であり、

図 3 は、本発明のマッチドフィルタにおける演算（コードスキャン・データ固
定方式）により、相関値を算出するまでの過程を説明するための図であり、

図 4 は、本発明のマッチドフィルタにおける演算方法の特徴的な処理手順を説
明するためのフロー図であり、

15 図 5 は、本発明のマッチドフィルタの、データ固定・コードスキャンによる演
算の進め方を説明するための図であり、

図 6 は、マッチドフィルタのデータ固定・コードスキャンによる演算の基礎的
な考え方を説明するための図であり、

図 7 は、本発明のマッチドフィルタのコードスキャン内容を説明する図であり、

20 図 8 は、本発明のデータ固定・コードスキャンによる演算を行うマッチドフィ
ルタの基本構成を示すブロック図であり、

図 9 は、図 8 のマッチドフィルタの積分区間選択回路の動作の概要を説明する
ための図であり、

図 1 0 は、本発明のマッチドフィルタを搭載した C D M A 受信機の構成を示す
25 ブロック図であり、

図 1 1 は、本発明の実施の形態 1 にかかるマッチドフィルタのブロック図であ

り、

図 1 2 は、図 1 1 のマッチドフィルタの動作を示すタイミングチャートであり、

図 1 3 は、本発明の実施の形態 2 にかかるマッチドフィルタのブロック図であ

り、

5 図 1 4 は、図 1 3 のマッチドフィルタの動作を示すタイミングチャートであり、

図 1 5 は、本発明の実施の形態 3 にかかるマッチドフィルタのブロック図であ

り、

図 1 6 は、図 1 5 のマッチドフィルタの動作を示すタイミングチャートであり、

図 1 7 は、本発明の実施の形態 4 にかかるマッチドフィルタのブロック図であ

10 り、

図 1 8 は、図 1 7 のマッチドフィルタの動作を示すタイミングチャートであり、

図 1 9 は、本発明の実施の形態 5 にかかるマッチドフィルタのブロック図であ

り、

図 2 0 は、図 1 9 のマッチドフィルタの動作を示すタイミングチャートであり、

15 図 2 1 は、従来のマッチドフィルタの基本的な構成を示すブロック図であり、

図 2 2 A は、マッチドフィルタを 2 つ使用して相関検出を行う、従来の構成を示したブロック図であり、

図 2 2 B は、図 2 2 A に示す従来構成の動作の概要を説明するためのタイミングチャートである。

20

(発明を実施するための最良の形態)

具体的な実施の形態について説明する前に、本発明の基本的な考え方について、図 1 ～図 7 を用いて説明する。

25 従来は、図 1 A に示すように、拡散コードを固定し、データをスキャンする方式であった。これに対し、本発明では、図 1 B に示すように、データを固定

し、拡散コードを1チップずつスキャンする方式を採用する。

図1Bに示されるように、入力データは、1シンボル期間(=Ts)に相当する処理期間Tnに区分され、これを単位としてシフトレジスタにラッチされ、固定される。図中、処理区間T1が現在の期間であり、T0は過去の処理期間であり、T2は未来の処理期間である。

処理期間(Tn)と、シンボル期間(Ts)とは独立した概念であり、同期がとれていない。つまり、シンボル期間(Ts)は、実際の受信信号の1シンボルに正確に対応する期間であり、これに対し、処理期間(Tn)は、シフトレジスタにラッチされるデータ長に対応する、処理を遂行ための便宜上の区分である。

よって、例えば、処理期間T1の中には、シンボル(n-1)の一部と、シンボル(n)の一部とが混在する。図1B中、このデータが混在する部分を点線で囲んで、DS1という符号で示している。

一つの処理期間中に、どのような状態で、隣り合う2つのシンボルが混在しているかは分からない。

そこで、シンボルとシンボルの境(シンボルの区切り)が存在するであろう全ての場合を想定し、それぞれの場合について積分区間をずらしながらシンボルを意識した積分(1シンボルの一部についての積分)を行う。そして、後に、積分結果を合体させて1シンボル全体についての相関値を算出し、その相関値を比較して相関のピークを求める。その相関のピークを与えるタイミングが、実際の受信データの入力タイミングである。

ここで、シンボルの区切りは、拡散コードを初期状態(基準のタイミング)から1チップずつずらしながらスキャンしていった、何チップ分だけ遅延したかに応じて、一義的に(自動的に)決まる。図1では、処理期間T1における、シンボルの、実際の区切りを太線で示している。

図示されるように、このシンボルの区切りを意識すると、処理期間T1では、

シンボル ($n - 1$) に属するデータについての逆拡散結果の積分値 (部分的な積分値) b と、シンボル (n) に属するデータについての逆拡散結果の積分値 (部分的な積分値) c とが求まることになる。

- 5 シンボル ($n - 1$) に属するデータについての逆拡散結果の積分値 b については、一つ前の処理サイクルの処理にて求められてメモリ 10 に蓄積されている、シンボル ($n - 1$) に属するデータについての逆拡散結果の積分値 a (この a は、同じ受信信号の遅延量に対応した積分値である) と加算される。

これにより、シンボル ($n - 1$) についての、一つの遅延量に対応した相関値が求まる。

- 10 また、シンボル (n) に属するデータについての逆拡散結果の積分値 c については、メモリ 12 に蓄積される。

そして、この積分値 c は、次の処理サイクルで得られる、シンボル (n) に属するデータについての逆拡散結果の積分値 d (この d は、同じ受信信号の遅延量に対応した積分値である) と加算される。これにより、シンボル (n) に

- 15 ついての、一つの遅延量に対応した相関値が求まる。

このようにして、一つのシンボルについての相関値 (と推定される値) が、1 チップ毎に求められ、その中の最大値を検出することにより、実際のそのシンボルについての相関値のピーク (シンボルのタイミング) が、検出されることになる。

- 20 図 2 に具体例を示す。

基地局の基準信号に対する受信データ ($A \sim I$) の遅延が、無しの場合と、1 チップ遅延している場合と、2 チップ遅延している場合を考える。いずれにしろ、処理期間 (現在の処理期間) T_0 で区切られて、データはシフトレジスタにラッチされる。

- 25 もし、遅延が無しならば、ラッチされるデータは “D, E, F” であり、この場合、処理期間とシンボルの区切り (シンボル期間) とは一致している。

一方、1チップ遅延の場合は、ラッチされるデータは“C, D, E”であり、シンボルの区切りは、データCとデータDとの間にある。

また、2チップ遅延の場合は、ラッチされるデータは“B, C, D”であり、シンボルの区切りは、データCとデータDとの間にある

- 5 ラッチされたデータ（DEFか、CDEか、BCDのいずれかである）について、拡散コード（逆拡散コード）を1チップずつずらしながら乗算する。

まず、遅延無しに対応する拡散コードが乗算される。この場合の拡散コードは“o, p, q”である。これらを、受信データの各々に乗算した結果を積分する区間はTS1（処理期間と一致する）である。

- 10 次に、乗算されるコードは、“n, o, p”である。これらを、受信データの各々に乗算した結果を積分する区間は、シンボルの区切りを意識して、TS2とTS3に分割される。

次に、乗算されるコードは、“m, n, o”である。これらを、受信データの各々に乗算した結果を積分する区間は、シンボルの区切りを意識して、TS

- 15 4とTS5に分割される。

図3の上側には、以上説明した処理が示されている。そして、図3の下側に示されるように、1チップ遅延の場合および2チップ遅延の場合において、部分的に積分して得られた積分値（b, c）は、それぞれ、一つ前のサイクルで求められた、同じ遅延に対応する積分値a、あるいは次のサイクルで求められる、同じ遅延に対応する積分値dと加算される。これにより、シンボル単位の相関値（と推定される値）S0～S4が求まる。S0は、遅延無しの場合の1シンボルの相関値であり、S1, S2はそれぞれ、1チップ遅延の場合の隣接するシンボルの部分的な相関値であり、S3, S4はそれぞれ、2チップ遅延の場合の隣接するシンボルの部分的な相関値である。

- 25 次に、これらのシンボル単位の相関値S0～S4を比較する。

図3の一番下にされるように、1チップ遅延の場合に、相関のピークが検出

される。したがって、ラッチされた受信データは1チップ分の遅延をもっていたことが判明する。すなわち、図2において、ラッチされた受信データは、“D, E, F”であったことがわかる。これにより、受信信号を処理するタイミングが検出されたことになる。

- 5 以上の方法によれば、入力データは固定なので、データをシフトレジスタに
入力するスピード（受信データの転送スピード）の限界は、なんら問題となら
ない。そして、基地局がカバーする伝播遅延に相当する期間に渡って、拡散コ
ードを1チップずつシフトして繰り返し乗算を行うので、ハードウェアがもつ
最大のスピードで演算処理を行える。ゆえに、相関値を求めるためのサーチの
10 時間幅（伝播遅延）が1シンボルに相当する時間を越えてた場合でも、マッチ
ドフィルタ1つで対応できる。

以上説明した本発明の特徴的動作をまとめると、図4のようになる。

すなわち、処理期間に対応したデータを切り出して、マッチドフィルタのメモ
リに蓄積する（ステップ20）。

- 15 1チップずつ位相をシフトしながら逆拡散を行い、コードのシフトしたチッ
プ数に対応させて積分期間を順次、切り替えつつ、逆拡散結果を積分する（ス
テップ21）。

- 次に、基地局の基準タイミングに対して、受信信号の遅延が無い場合を想定
している場合には（ステップ22）、逆拡散結果の積分値がそのまま、1シン
20 ボルについての相関値とする（ステップ23）。

- 一方、ステップ22で、受信信号に遅延がある場合を想定している場合には、
一つのデータの区切りにおいて、異なるシンボルのデータが混在することにな
るので、前半のシンボルについての積分結果は、すでに算出されている過去の
（同じ遅延に対応する）積分結果と加算してそのシンボルについての相関を検
25 出する。また、後半のシンボルについては、その積分結果をメモリに一時的に
蓄積する。蓄積された積分結果は、次のサイクルで算出される、対応するシン

ボルの一部についての積分結果と加算され、そのシンボルについての相関が検出される（ステップ24）。

そして、シンボル毎の相関値を比較してピーク値を求め、受信データの、実際の遅延量を検出する（ステップ25）。

- 5 以上が、本発明のデータ固定・コードスキャン方式の相関検出の基本的な考え方である。

図5に、サーチ幅を2シンボルに相当する時間に設定した場合における、処理期間と、受信信号のシンボルの区切りとの相対的な位置関係を示す。処理期間（ $T_1 \sim T_{n+2}$ ）のそれぞれは、1シンボルに相当する時間幅である。

- 10 図6は、図4の一部を拡大して示す図である。図中、 $A_n(T)$ や $B_n(T)$ は、逆拡散結果の部分的な積分値を示す。

- $n-1$ 番目の処理期間（ T_{n-1} ）、 n 番目の処理期間（ T_n ）、 $n+1$ 番目の処理期間（ T_{n+1} ）のいずれの場合も、シンボルの区切れ P_1 , P_2 , P_3 と、処理期間の始端と終端（ t_{n-2} , t_{n-1} , t_n , t_{n+1} ）との
- 15 相対的な位置関係は同じである。図6では、一つの処理期間（例えば、 T_n ）は、シンボルの区切れ（例えば、 P_2 ）により2分割されるが、この2分割された各期間の長さは、同じ伝播遅延量の受信信号については、どの処理期間でも同じである。よって、各処理期間において、2分割されたそれぞれの区間について独立に逆拡散と積分を実行して部分的な積分値を求め、次の処理期間に
- 20 において求められる部分的な積分値と合算しても、何も問題は生じない。本発明では、このような考え方にに基づき、各処理期間毎に部分的積分処理を行い、次の処理の際に、部分的な積分値同士を加算する。

ここで、伝播遅延が“ T ”の場合に着目する。

- 処理期間 T_{n-1} では、部分的な積分値 $B_{n-1}(T)$ が算出され、メモリ
- 25 （不図示）に記憶される。

そして、次の処理期間 T_n では、部分的な積分値 $A_n(T)$ と $B_n(T)$ が

求められ、 $A_n(T)$ が、メモリから読み出される部分的な積分値 $B_{n-1}(T)$ と加算される。これにより、1シンボル分の相関値が求められる。 $B_n(T)$ は、メモリに格納され、次の処理で求められる部分的な積分値 $A_{n+1}(T)$ に加算されて1シンボル分の相関値が求められる。

- 5 このように、本発明の、データ固定・コードスキャン方式によって求められる遅延プロファイルは、従来のデータスキャン・コード固定方式によって求められるものと全く同じである。

ここで、本発明のデータ固定・コードスキャンによる相関検出のための演算を行う場合に発生させる拡散コードを、図7を用いて説明する。

- 10 図示されるように、伝搬遅延0チップの受信信号 $DA(n) \sim DA(n+255)$ に対しては、拡散コード $CO(n) \sim CO(n+255)$ を乗算する。同様に、伝搬遅延が1チップの受信信号 $DA(n-1) \sim DA(n+254)$ に対しては、拡散コード $CO(n-1) \sim CO(n+254)$ を乗算する。

- ここで、受信信号の伝搬遅延量の変化に伴う拡散コードの変化に着目すると、
15 伝搬遅延が1チップ増加する毎に、拡散コードも1チップずつ後ろにずれていくことがわかる。

したがって、伝搬遅延が0チップである場合の拡散コードを初期値とし、その後、1チップずつずらした内容の拡散コードを次々に生成する。次々に発生する拡散コードを、固定されたデータに対して乗算する。

- 20 以下、本発明の実施の形態について、図面を参照して説明する。

(実施の形態1)

図8に、本発明のマッチドフィルタの基本的な構成例を示す。

- 本発明のマッチドフィルタ回路201は、受信データをシリアルに入力し、
25 パラレルに出力するデータレジスタ202と、逆拡散演算部203と、積分演算部204と、拡散コード（逆拡散コード）を発生させるコード発生器206をもつ。

データレジスタ 202 には、受信したデータがシリアルにされる。そして、先に図 5, 図 6 を用いて説明した、一つの処理期間 (T_n) に対応する 1 シンボル分のデータが蓄積されると、受信データをパラレルに出力する。すなわち、データレジスタ 202 は、シリアル/パラレル変換回路として機能する。

- 5 データレジスタ 202 からパラレルに出力されたデータは、データレジスタ 205 にラッチされる。

コード発生器 206 は、排他的論理和回路 214, 215 と、シフトレジスタ 213 を組み合わせて構成される、巡回型シフトレジスタをもつ。この巡回シフトレジスタは、M 系列拡散符号の発生器である。このコード発生器 206
10 からは、1 チップずつシフトした拡散符号が発生する。本実施例では、1 シンボル (= 256 チップ) に対応する拡散符号をパラレルに出力する。

このコード発生器 206 から発生される 256 ビットの拡散コードは、コードレジスタ 207 に蓄積される。

拡散回路 208 は、データレジスタ 205 からパラレルに出力される 1 シン
15 ボル相当の受信データに、コードレジスタ 207 からパラレルに出力される拡散コードを乗算する。

積分演算部 204 は、積分区間選択回路 209 と、積分回路 210 と、メモリ 211 と、加算回路 212 をもつ。

積分区間選択回路 209 には、逆拡散回路 208 から出力される 256 ビッ
20 トの逆拡散演算結果を示すデータビットがパラレルにされる。積分区間選択回路は、256 ビットのデータを任意の位置で 2 つに分割し、一方のグループのデータビットのみをパラレルに出力することができる。つまり、256 ビットのデータの中から、積分回路 210 に送るデータビットを選択することができる。これにより、図 6 で説明した、各処理期間における、部分的な積分が
25 実現される。具体的には、積分区間選択回路 209 は、受信データの想定される遅延量に連動して、自動的に積分区間を連続的に切り替えていく。積分区間

の区切り目（256ビットのデータを2分割する位置）は、図6で説明したように、常に、実際の受信データにおける1つのシンボルの区切りとなるであろう位置（図6のP1, P2, P3）である。

積分回路10において、積分演算が実行され、これにより、図6に示される
5 部分積分値（現在の処理期間が T_n であれば、 $A_n(T)$ や $B_n(T)$ ）が得られる。

得られた部分積分値のうち的一方（例えば、図6の $B_n(T)$ ）は、メモリ211にライトされる。メモリ211は、伝搬遅延に相当するアドレス空間を有する。積分回路210から出力される他方の部分積分値（例えば、図6の
10 $A_n(T)$ ）は、メモリ211からリードされる、先の処理において得られた部分的な積分値（例えば、図6の $B_{n-1}(T)$ ）と、加算回路212において、加算される。これにより、1シンボルについての相関値が求められる。

求められた相関値（相関出力）は、一旦、メモリ215に蓄積される。

そして、ピーク検出回路216により、蓄積された相関値どうしを比較して
15 相関のピークを求める。その相関のピークに対応して、受信データの遅延量が定まる。これにより、受信信号を処理するためのタイミングが検出される（図3）。

図9は、積分区間選択回路209の動作を説明するための図である。

上述したように、積分区間選択回路209は、受信データの想定される遅延
20 量に連動して、自動的に積分区間を連続的に切り替える。積分区間の区切り目（256ビットのデータを2分割する位置）は、図6で説明したように、常に、実際の受信データにおける1つのシンボルの区切りとなるであろう位置（図6, 図9のP1, P2, P3）である。

したがって、図8に示すように、受信信号の伝播遅延が0～1チップまでの
25 範囲では、シンボルの区切れP1にて、256ビットのデータが2分割され、これにより、積分期間は自動的に決まる。同様に、伝播遅延が1チップ～2チ

ップの範囲では、シンボルの区切れP 2に応じて積分期間が決まり、伝播遅延が2チップ～2.5チップの範囲では、シンボルの区切れP 3に応じて積分期間が決まる。

このようにして、積分期間選択回路209により、受信信号におけるシンボルの区切りに対応して、積分区間が連続的に切り替えられる。

(実施の形態2)

図10は、本発明のマッチドフィルタを具備するCDMA受信装置の要部構成を示すブロック図である。

この受信機は、受信アンテナ401と、所定のフィルタリング及び増幅する高周波信号処理部402と、A/D変換器403と、復調器409と、復号器410と、復号された信号を音声に換えるCODEC部411と、マッチドフィルタ404と、クロック発生器407と、PLL回路408とをもつ。

スペクトラム拡散された受信信号は、マッチドフィルタ404のデータレジスタ405でシリアル/パラレル変換され、相関演算部406のデータレジスタに格納され、固定される。

マッチドフィルタ404では、固定されたデータに対して、PLL回路408から供給される高速なクロック信号に基づいて、高速にコードを乗算することができる。したがって、非常に高速な逆拡散演算処理が実現する。

これにより、本発明のマッチドフィルタを具備するCDMA受信装置は、データ固定・コードスキャンの演算を高速に行うことにより、システムが要求するサーチ範囲が1シンボルを超える時間幅であっても、1つのマッチドフィルタ404にて対応可能である。よって、本発明のCDMA受信機を用いることにより、基地局のセル半径の拡大を容易に実現できる。

また、本発明のマッチドフィルタは、きわめて高速に演算を行えるため、異なる条件の遅延プロファイル演算を、例えば、時分割で行うことができる。つまり、少ないハードウェアで、高度な処理を高速に行うことができる。よって、

低消費電力で、かつ高機能な携帯電話機が実現可能である。

(実施の形態 3)

図 11 は、本発明のマッチドフィルタの他の構成例を示す図である。

このマッチドフィルタ回路 301 は、データレジスタ 202 と、コード発生
5 器 206 と、逆拡散演算部 203 と、積分演算部 302 をもつ。

積分演算部 302 は、積分区間選択回路 304 と、それをコントロールする
信号を生成するデコーダ部 303 と、選択された積分区間の結果を演算する加
算器 305 と、逆拡散演算部 203 から出力される出力全てを加算する加算器
307 と、加算器 307 の結果から加算器 305 の結果を減算する演算器 30
10 8 と、加算器 305 の結果を蓄積するメモリ 306 と、演算器 308 の出力と
メモリ 306 から読み出される内容を加算する加算器 309 と、をもつ。

逆拡散演算部 203 のデータの読み込みは DATA LOAD 信号に基づいて
行われる。積分区間選択回路 304 の積分区間の指定は、デコーダ部 303 か
ら出力される DECDIV 信号に基づいて行われる。デコーダ部 303 の DE
15 CDIV 信号の生成は、DIV 信号に基づいて行われる。

メモリ 306 からデータを読み出すときは、アドレスを ADDRESS 信号
により指定し、READ 信号を入力する。また、メモリ 306 にデータをライ
トするときは、WRITE 信号を入力する。メモリ 306 は、同一のクロック
で、リードおよびライトを同時に行うことができる。

20 図 11 のマッチドフィルタでは、1つの処理期間（例えば、図 6 の T_n ）に
おける 1 シンボル分の逆拡散結果を、加算器 307 にて、すべて加算する（図
6 における $A_n(T) + B_n(T)$ ）。

また、積分区間選択回路 304 において、シンボルの区切り（図 6 の P_2 ）
よりも後に位置する部分のデータを部分的に積分する（図 6 の $B_n(T)$ ）。

25 そして、演算器 308 において、加算器 307 の出力（図 6 の $A_n(T) +$
 $B_n(T)$ ）から、加算器 305 の出力（図 6 の $B_n(T)$ ）を減算する。一

方、メモリ306に、加算器305の出力($B_n(T)$)をライトし、同時に、メモリ306から、直前の処理にて算出された部分的な積分値(図6の $B_{n-1}(T)$)をリードする。

そして、加算器309において、メモリ306から出力される部分的な積分値(図6の $B_{n-1}(T)$)と、加算器308の出力(図6の $A_n(T)$)とを加算する。これにより、1シンボルについての相関値(図6の $B_{n-1}(T) + A_n(T)$)が得られる。

次に、図11のデータレジスタ202および逆拡散演算部203と積分演算部302の、より具体的な動作について説明する。

10 図12は、本発明の実施の形態1のマッチドフィルタの演算回路の動作を示すタイミングチャートである。図12に記載されているA~Dは、図11における信号A~Dを表している。

データレジスタ202には、常に、受信データがシリアル入力されている。

まず、時刻T1において、DATALOAD信号を有効化し、クロック信号
15 に同期した時刻T2において、データレジスタ202の内容を逆拡散演算部203に取り込む。

時刻T2のタイミングで、演算対象の伝搬遅延0で届く受信シンボルを逆拡散するための拡散コードを逆拡散演算部203に格納し、逆拡散演算を行う。

さらに同じ時刻T2のタイミングで、加算器307は、逆拡散演算部203
20 の出力全てを加算し、図12のC2(0)として出力する。

また、同じ時刻T2のタイミングで、デコード回路303に与えられるDIV信号が“0”となり、その結果、256ビットのDECDIV信号の各ビットが全てハイレベルに固定され、これが積分区間選択回路304に与えられる。

積分区間選択回路304では、DECDIV信号の全ビットがハイレベルであるため、出力を全てロウレベルに固定する。加算器305は、“0”を、図
25 12におけるB2(0)として出力する。

同じタイミングで、メモリ306の、ADDRESS信号で示されるアドレスから1つ前の処理期間における値“0”をB1(0)として読み出す。

その直後、メモリ306の同じアドレスにB2(0)を書き込む。

その結果、演算器308は、C2(0)からB2(0)を減算した値、すな
 5 わちA2(0)を出力し、加算器309において、メモリ306から読み出したB1(0)と加算器309の出力A2(0)とを加算し、図12の相関演算結果D2(0)として出力する。

時刻T3以降の演算内容は、上で説明した内容の繰り返しである。

時刻T2から時刻T4までのDIV信号は、受信対象シンボルの想定される
 10 伝搬遅延に連動して変化し、“0”～“255”が繰り返される。

図12の、DECDIV1～DECDIV255は、DIV信号に基づいて図12に示すようなパターンで周期的に変化する。

また、メモリ306は、受信対象シンボルの想定される伝搬遅延に対応したデータ格納領域を有している。受信信号の伝搬遅延に応じて、アクセス対象の
 15 領域が切り替えられる。

図12における、WRITEおよびREAD信号については、メモリ306が同一クロックでのリードおよびライト動作が可能であるため、演算中は、常にアクティブとなっている。

図12のタイミングチャートは、256チップからなる一つの処理期間において、511チップ相当のサーチ幅の遅延プロファイルを生成する場合の例を示したものである。演算処理に用いられるクロックのスピードを上げれば、さらに大きなサーチ幅にも対応可能である。

このように、本実施例のマッチドフィルタの演算回路によれば、データ固定・コードスキャン方式による相関演算の結果を、1クロック毎に出力することが
 25 できる。

(実施の形態4)

図13は、本発明のマッチドフィルタの、他の実施例の構成を示す図である。

本実施例のマッチドフィルタの機能と動作は、図11のマッチドフィルタと同様である。ただし、本実施例では、逆拡散後のデータの累積加算を先行して行う。これにより、積分区間選択回路をセレクタで構成することができ、回路
5 の簡素化を図ることができる。

図13のマッチドフィルタ回路501は、データレジスタ202と、コード発生器206と、逆拡散演算部203と、積分演算部502をもつ。

積分演算部502は、累積加算部503と、積分区間選択回路505と、メモリ306と、演算器506、507をもつ。

10 累積加算部03は、複数の加算要素504a～505eをもつ。この複数の加算要素504a～505eは、逆拡散演算部203から出力される256ビットのデータに関して、下位ビットから順に累積的に加算していく働きをする。図13中、b1～b255はそれぞれ、累積加算の結果を示すデータである。

15 積分区間区間選択回路505は、累積加算部503から出力される255個の累積加算結果を示すデータの中から、いずれか一つ（図中のデータB）を選択する。データBは、メモリ306にライトされると共に、演算器506に送られる。

20 加算器505eからは、逆拡散結果を示すデータビットの全てを加算した結果（図中のデータC）が得られる。演算器506では、 $(C-B)$ の演算がなされ、その結果、データAが得られる。このデータAは、メモリ306からリードされるデータと加算され、相関出力Dが得られる。

逆拡散演算部203のデータの読み込みはDATA LOAD信号に基づいて行われる。積分区間選択回路505の積分区間の指定はDIV信号に基づいて行われる。

25 データレジスタ202、コード発生器206、逆拡散演算部203、メモリ306、演算器506、加算器507については、前掲の実施例と同じである

ため説明を省く。

図14は、図13に示すマッチドフィルタの動作を示すタイミングチャートである。

まず、時刻T1においてDATALOAD信号を有効化し、クロック信号に
5 同期した時刻T2のタイミングでデータレジスタ202の内容を逆拡散演算部
203に取り込む。

また、同じく時刻T2のタイミングで、加算器ブロック503は、逆拡散演
算部203の1タップ毎の出力を次々と累積加算して行き、そのタップ毎の加
算結果としてb2～b255を出力し、また全タップの加算結果を図14のC
10 2(0)として出力する。

同じく、時刻T2のタイミングで、DIV信号が“0”となり、その結果、
積分区間選択回路505からは、図14のB2(0)として出力される。

同じタイミングで、メモリ306から1つ前の処理において取得されたデー
タ“0”を、図14のB1(0)として読み出す。

15 その結果、演算器506は、C2(0)からB2(0)を減算した値、すな
わちA2(0)が出力される。そして、加算器507において、メモリ306
から読み出したB1(0)と加算器309の出力A2(0)とが加算され、図
14の相関演算結果D2(0)が得られる。

時刻T3以降の演算内容は、上で説明した内容の繰り返しである。

20 このように、本実施例のマッチドフィルタによれば、1クロック毎にデータ
固定・コードスキンの相関演算結果を出力することができる。また、本実施
例のマッチドフィルタでは、積分区間選択回路をセレクタで構成することがで
き、回路の簡素化を図ることができる。

(実施の形態5)

25 図15は、本発明のマッチドフィルタの他の例の構成を示すブロック図であ
る。

マッチドフィルタの基本的な機能や動作は、前掲の実施例と同じである。ただし、本実施例では、一つの積分区間選択回路を時分割で使用することで、異なる2つの受信データを交互に処理する。処理したい信号の数よりも、使用できる回路が少ない場合に、一つの回路を共用する場合に適した構成である。

- 5 前掲の実施例では、1クロックに1回、相関値が出力されたのに対し、本実施例では、2クロックに1回、一つの受信信号についての相関値が出力される。

本実施例では、マッチドフィルタに入力される信号は、2種類の受信信号のデータを1チップずつ交互に配置して、合成された信号である。つまり、2種類の信号をQ, Wとすると、入力信号は、Q1, W1, Q2, W2・・・というように、1チップずつ交互に配置される。どちらの信号の相関値を求めるかは、MASKEN信号の周期を切り替えることにより、選択することができる。

このマッチドフィルタ回路601は、データレジスタ202と、コード発生器206と、逆拡散演算部203と、積分演算部602をもつ。

積分演算部602は、積分区間選択回路304と、デコーダ部303と、加算器305と、レジスタ603と、加算器305の加算結果からレジスタ603から出力されるデータ値を減算する演算器605と、メモリ604と、演算器605の出力とメモリ604から読み出されるデータを加算する加算器606とをもつ。

逆拡散演算部203のデータの読み込みは、DATALOAD信号に基づいて行われる。積分区間選択回路304の積分区間の指定は、デコーダ部303から出力されるDECDIV信号およびMASKEN信号に基づいて行われる、デコーダ部303から出力されるDECDIV信号の内容は、DIV信号に基づいて定まる。

メモリ604からのリードおよびメモリ604へのライトは別クロックで行う。

図16は、図15のマッチドフィルタの動作を示すタイミングチャートであ

る。

データレジスタ202には、常に、受信データがシリアルに入力される。

- 時刻T1においてDATA LOAD信号が有効化され、クロック信号に同期した時刻T2のタイミングで、データレジスタ202に保持されているデータ
5 を逆拡散演算部203に取り込む。

時刻T2において、演算対象の伝搬遅延“0”の受信シンボルを逆拡散するための拡散コードを逆拡散演算部203にセットし、逆拡散演算を行い、その結果を、時刻T4まで保持する。

- 同じく、時刻T2のタイミングで、デコード回路303へのDIV信号が“0”
10 となり、また、積分区間選択回路203に与えられるMASKEN信号が無効化される。これにより、逆拡散演算部203の出力は、積分区間選択回路203を通過して、そのまま出力される。

その結果、加算器305の出力は逆拡散演算部203の出力ビットの全てを加算した出力となる。これが、図16におけるデータC2(0)である。

- 15 次に、時刻T3において、レジスタ603に対してC2(0)が書き込まれる。

時刻T3において、MASKEN信号が有効化される。また、時刻T2において、DIV信号により、DEC DIV信号の全てのビットがハイレベルとなる。よって、積分区間選択回路304の出力が全てロウレベルに固定される。

- 20 加算器305による加算結果は“0”となる。この加算結果が、図16のB2(0)である。

演算器605からは、C2(0)からB2(0)を減算した結果A2(0)が出力される。

- 時刻T2で、READ信号が有効となる。そして、時刻T3において、メモリ604からデータB1(0)が出力される。
25

加算器606からは、A2(0)とB1(0)を加算した相関演算結果D2

(0) が出力される。

時刻 T 4 において、メモリ 6 0 4 の同じアドレスにデータ C 2 (0) を書き込む。

時刻 T 4 以降の演算内容は、上で説明した内容の繰り返しである。

- 5 時刻 T 2 から時刻 T 6 までの D I V 信号は、前掲の実施例と同じ働きをするが、2クロック周期であるところが前掲の実施例とは異なっている。

DECDIV1~DECDIV255は、DIV信号に基づいて、図16に示すようなパターンで周期的に変化する。

- 10 本実施例のマッチドフィルタによれば、データ固定・コードスキャン方式の
 10 相関演算結果を、2クロック毎に出力することができる。また、積分区間選択回路を時分割使用することにより、限られたハードウェア資源を活用して、多くの信号を処理することができる。

(実施の形態6)

- 15 図17は、本発明のマッチドフィルタの他の例の構成を示すブロック図である。

本実施例のマッチドフィルタの機能は、図15のマッチドフィルタと同じである。つまり、積分区間選択回路を時分割で使用し、2クロック毎に相関値を出力する。

- 20 ただし、図15の実施例では、積分区間選択回路304における積分区間を
 20 デコーダ303で制御していたが、本実施例では、シフトレジスタ704を用いて制御する。これにより、回路構成を簡素化することができる。

シフトレジスタ704は、初期値生成回路703が生成する初期値信号と、LOAD信号およびSHIFTEN信号が与えられる。

- 25 初期値生成部703とシフトレジスタ部704の具体的な動作を図18を用
 25 いて説明する。

データレジスタ202には、常に受信データがシリアル入力される。

まず、時刻T1においてDATALOAD信号が有効化される。クロック信号に同期した時刻T2のタイミングでデータレジスタ202に保持されているデータを逆拡散演算部203に取り込む。

- 5 時刻T2のタイミングで、演算対象の伝搬遅延“0”の受信シンボルを逆拡散するための拡散コードを逆拡散演算部203に格納し、逆拡散演算を行い、その結果を、時刻T4まで保持する。

時刻T1において、LOAD信号を有効化される。クロック信号に同期した時刻T2のタイミングで、初期値生成回路703が生成する初期値を、シフトレジスタ部704に取り込む。

- 10 時刻T2のタイミングで、SHIFTEN信号が無効化される。これにより、時刻T3に、シフトレジスタ704のシフト動作が停止される。

同じく、時刻T2のタイミングで、積分区間選択回路304に入力されるMASKEN信号が無効化される。これにより、逆拡散演算部203から出力されるデータとまったく同じデータが、積分区間選択回路304から出力される。

- 15 時刻T3のタイミングでSHIFTEN信号が有効化される。クロックに同期した時刻T4のタイミングで、シフトレジスタのシフト動作が有効となる。このようにして、2クロックに1回のシフト動作が実現される。

- このように、本実施例では、積分区間選択回路を制御するデコード回路の代わりにシフトレジスタ回路を用い、積分区間選択回路の制御回路を簡素化することができ。実現されるマッチドフィルタの機能は、図15のマッチドフィルタと同じである。
- 20

(実施の形態7)

図19は、本発明のマッチドフィルタの他の例を示すブロック図である。

- 本実施例のマッチドフィルタの機能は、図17のマッチドフィルタと同じである。ただし、図17のマッチドフィルタでは、相関演算結果Dを求めるために、逆拡散演算された全ビットの積分値Cから、シフトレジスタ704で指定
- 25

された積分結果Bを減算して積分結果Aを算出しているため、回路構成がやや複雑である。

そこで、本実施例では、積分区間を選択するシフトレジスタ704の出力を反転する反転制御回路803を設け、シフトレジスタ704の出力ビットを部分的に反転させることにより、早い段階で、積分値Aを算出する。これにより、加算器やメモリ等をかなり削減でき、回路の簡素化を図ることができる。

図19から明らかなように、反転制御回路803を設けることにより、相関値を演算する最終段の回路の構成要素が、メモリ306と加算器309のみになり、回路構成が簡素化されている。

10 積分区間選択回路304の積分区間は、反転制御回路803の出力により決定される。反転制御回路803の出力データは、INV信号に基づき、シフトレジスタ704の出力ビットを反転させることにより得られる。

反転制御回路803の具体的な動作について、図20を用いて説明する。

データレジスタ202には、常に受信データがシリアルに入力されている。

15 まず、時刻T1においてDATALOAD信号が有効化される。クロック信号に同期した時刻T2のタイミングで、データレジスタ202に保持されているデータが逆拡散演算部203に取り込まれる。

時刻T2のタイミングで、演算対象の伝搬遅延“0”の受信シンボルを逆拡散する拡散コードを逆拡散演算部203にセットし、逆拡散演算を行い、その

20 結果を時刻T4まで保持する。

時刻T1において、LOAD信号を有効化する。クロック信号に同期した時刻T2のタイミングで、初期値生成回路703の出力データを、シフトレジスタ部704に取り込む。

時刻T2のタイミングで無効化されたSHIFTEN信号は、時刻T3のタイミングで、シフトレジスタ704のシフト動作を無効とする。

時刻T2のタイミングで、反転制御回路803に入力されるINV信号のレ

レベルをロウレベルに固定する。これにより、シフトレジスタ704から出力される全ビットが“1”であるデータが、そのまま積分区間選択回路203に対して与えられる。積分区間選択回路203からは、逆拡散演算部203の全タ
 5 ュップの出力ビットと同じデータビットが出力される。加算器305は、これらのデータビットを加算する。この加算結果が、図20におけるA2(0)である。

時刻T3においては、INV信号をハイレベルに固定する。これにより、シフトレジスタ704が出力するオール“1”の信号が全ビットについて反転され、オール“0”のデータビットとなる。このオール“0”の信号が、積分区
 10 間選択回路304に与えられ、積分区間選択回路304からは逆拡散演算部203の全データビットの値がマスクされた、すべてがロウレベルであるデータ（オール“0”のデータ）が出力される。加算器305からは、図20のB2(0)として、“0”が出力される。

時刻T3において、メモリ306のADDRESS信号にて指定されるアド
 15 レスから、データB1(0)がリードされる。

その結果、加算器309からは、A2(0)とB1(0)を加算した相関演算結果D2(0)が出力される。

次に、図18の時刻T4において、メモリ306の同じアドレスに、データB2(0)をライトする。

20 時刻T4以降の演算内容は、上で説明した内容の繰り返しである。

実施の形態5では、相関演算結果Dを求めるために逆拡散演算された処理区切りの区間の積分値Cからシフトレジスタ704で指定された積分結果Bを減算しての積分結果Aを算出していた。これに対し、本実施例では、積分区間を選択するシフトレジスタ704の出力を反転し、これにより、積分値Aも算出
 25 できるようにした。

図20において、DECDIV1~DECDIV255は、DIVに基づい

て、図20に示すようなパターンでもって周期的に変化する。

このように、本実施例によれば、積分区間選択回路を制御するシフトレジスタの出力の反転・非反転を制御する反転制御回路を設けることにより、加算器や記憶回路の数を削減することができる。

- 5 以上、本発明をいくつかの実施の形態に基づいて説明した。上述の説明では、主に、CDMA通信の基地局に使用することを前提として説明したが、これに限定されるものではない。例えば、移動局側で、あるデータ列に対して異なる複数種類のコードとの相関を検出するような場合にも、本発明を容易に適用することができる。
- 10 以上説明したように、本発明によれば、データ固定・コードスキャンの相関演算を行うことができ、したがって、遅延プロファイル演算が従来にくらべて容易に高速化できる。すなわち、データの入力速度に起因する制約はなくなり、純粹にマッチドフィルタを構成するハードウェアの最大の処理能力で演算処理を行うことができる。
- 15 その結果、所定の演算をより小規模のマッチドフィルタ回路で実現することができる。よって、基地局のセル半径の大型化に容易に対応できる。また、移動局（携帯電話）において、異なる条件での遅延プロファイル演算処理を時分割で行うことができる。また、1シンボル以上の伝播遅延をカバーする場合も、1つのマッチドフィルタで対応できる。本発明によれば、回路規模が抑制され
- 20 るので、スペースの削減を図ることができる。よって、LSIの小型化、さらにはシステムの小型化を実現できる。

この出願は、2000年3月31日にファイルされた、日本国の特許出願2000-099306に基づいており、その全ての内容は、ここに明確に記載されている。

本発明は、CDMA受信機における相関検出器（マッチドフィルタ）に適用することができる。

請求の範囲

1. 最大で X チップ (n は自然数) 相当の遅延をもつ受信信号について、遅延プロファイルを作成することができる相関検出方法は、

- 5 前記受信信号のデータ列から、1シンボル相当のデータを切り出して固定する第1のステップと、

拡散コードの位相シフト量を0チップから X チップまで変化させることにより、1チップ単位の遅延をもつ拡散コードを連続的に生成し、生成された各拡散コードを前記固定されたデータに乗算して逆拡散結果を得ると共に、前記拡散コードの位相シフト量に応じて一義的に定まる受信信号のシンボルの仮想的な区切れを考慮して、積分区間を変更しながら、前記逆拡散結果についての積分を実行し、積分値を保存する第2のステップと、

10

- 前記固定された1シンボル相当のデータに隣接する、1シンボル相当のデータを新たに切り出して固定し、前記処理と同様の処理を実行する第3のステップと、
- 15 プと、

前記第2のステップおよび前記第3のステップで得られる、拡散コードの同じ位相シフト量に対応する積分値であって、前記受信信号のシンボルの仮想的な区切れを考慮した場合に、同じシンボルについての積分値であると考えられる積分値どうしを加算して、一つのシンボルに関する相関値を算出する第4の

20 ステップと、

算出されたシンボル単位の相関値どうしを比較し、最大の相関を示すものを検出することにより、前記受信信号の実際の遅延量を検出する第5のステップと、

を、含む。

- 25 2. 請求項1の相関検出方法において、

前記各ステップを一つのマッチドフィルタを用いて実行して、1シンボル相

当の時間よりも長い遅延をもつ受信信号の遅延プロファイルを作成する。

3. 相関検出方法は、

入力データを一時的に保持してデータを固定すると共に、その固定されたデータに対して、拡散コードの位相を変化させながら逆拡散を行うステップと、

- 5 前記拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前に位置する前半のシンボル部分と、後に位置する後半のシンボル部分の各部分について、逆拡散結果を積分するステップと、

- 前記前半のシンボル部分についての積分結果を、先に実施された同様の処理の結果として保持されている、同じシンボルについての、前記拡散コードの同じ位相シフト量に対応する積分結果と加算し、一方、前記後半のシンボル部分
10 についての積分結果は一時的に保持しておき、その保持されている積分結果を、次に同様の処理を実施した結果として得られる、同じシンボルについての、前記拡散コードの同じ位相シフト量に対応する積分結果に加算し、これによって、シンボル単位の相関を検出するステップと、

- 15 を、含む。

4. 請求項 3 の相関検出方法において、

検出されたシンボル単位の相関値どうしを比較し、最大の相関を示すものを検出することにより、前記入力データの実際の遅延量を検出するステップを、さらに有する。

- 20 5. 請求項 3 の相関検出方法において、

前記入力データとして、2 以上の種類の信号が 1 チップ毎に交互に配置されて多重化されたシリアルデータを採用すると共に、多重度に応じてデータ処理のタイミングを制御することにより、前記 2 以上の種類の信号のうちの、相関検出の対象となる信号についてのみ前述の各ステップの処理を行う。

- 25 6. マッチドフィルタは、

入力データを蓄積する一時的保持回路と、

位相が1チップ単位でシフトされる拡散符号を連続的に生成する拡散コード発生器と、

前記拡散符号を、前記一時的保持回路に保持された前記入力データに乗算する逆拡散演算回路と、

- 5 前記拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前に位置する前半のシンボル部分と、後に位置する後半のシンボル部分の各々の部分について、逆拡散結果を積分する積分回路と、

前記後半のシンボル部分の積分結果を一時的に蓄積する蓄積回路と、

- 10 前記前半のシンボル部分に関する積分結果を、先に実施された同様の処理の結果として前記蓄積手段に保持されている、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果に加算し、1つのシンボルについての相関値を出力する演算回路と、

を有する。

7. 請求項6のマッチドフィルタにおいて、

- 15 前記積分回路は、前記逆拡散演算回路の全出力ビットについて積分を行って得られる積分結果から、前記後の部分の積分結果を減算することにより、前記前の部分の積分結果を得る。

8. マッチドフィルタは、

入力データを蓄積する一時的保持回路と、

- 20 位相が1チップ単位でシフトされる拡散符号を連続的に生成する拡散コード発生器と、

前記拡散符号を、前記一時的保持回路に保持された前記入力データに乗算する逆拡散演算回路と、

- 前記逆拡散演算回路の出力の各ビットのデータを、最下位ビットあるいは最上位ビットを起点として順に累積加算していき、その結果として得られる複数の累積加算値を、並列に出力する累積加算演算部と、
- 25

前記累積加算演算部から出力される前記複数の累積加算値を選択するセクタと、

- 前記逆拡散演算回路の全出力ビットについて積分を行って得られる積分結果から、前記セクタにより選択された累積加算値を減算することにより、前記
- 5 拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前に位置する前半のシンボル部分についての逆拡散結果の積分値を算出する演算回路と、

- 前記前半のシンボル部分に関する前記積分値を、先に実施された同様の処理の結果として取得されて保持されている、同じシンボルについての、拡散コード
- 10 の同じ位相シフト量に対応する積分値に加算し、1つのシンボルについての相関値を出力する演算回路と、

を有する。

9. 2以上の種類の信号が1チップ毎に交互に配置されて多重化されたシリアルデータから所定の幅のデータを切り出して固定し、この固定されたデータに
- 15 対して、1チップ単位でシフト量が増加する拡散コードを連続的に乗算して逆拡散を行い、相関値を求めるマッチドフィルタは、

前記所定の幅のデータを蓄積する一時的保持回路と、

位相が1チップ単位でシフトされる拡散符号を連続的に生成する拡散コード発生器と、

- 20 前記拡散符号を、前記一時的保持回路に保持された前記入力データに乗算する逆拡散演算回路と、

- 前記固定されたデータの多重度に応じて、データ処理のタイミングを制御することにより、実質的に、前記2以上の種類の信号のうちの、相関検出の対象となる信号についてのみに信号処理を施し、これにより、その相関検出の対象
- 25 となる信号に関し、前記拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前に位置する前半のシンボル部分と、後に位置する後

半のシンボル部分の各々の部分について、逆拡散結果を積分して積分値を得る積分回路と、

前記後半のシンボル部分の積分結果を一時的に蓄積する蓄積回路と、

- 5 前記前半のシンボル部分に関する積分結果を、先に実施された同様の処理の結果として前記蓄積手段に保持されている、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果に加算し、1つのシンボルについての相関値を出力する演算回路と、
- を有する。

- 10 10. 請求項9のマッチドフィルタにおいて、
- 前記積分回路は、シフトレジスタを用いて、積分を行う範囲を制御する。
11. 請求項10のマッチドフィルタにおいて、
- 前記積分回路は、シフトレジスタと、前記シフトレジスタの出力ビットの反転・非反転を制御する反転／非反転制御回路と、を用いて積分を行う範囲を制御する。
- 15 12. 請求項6～請求項11のいずれかに記載のマッチドフィルタを具備し、前記マッチドフィルタの相関検出結果に基づいて、同期獲得処理または同期追従処理を行うCDMA受信装置。
13. 請求項12記載のマッチドフィルタを用いてスペクトラム拡散変調信号について同期を獲得し、獲得した同期タイミングに基づいて通信制御を行う
- 20 移動体通信基地局装置。
14. 請求項12記載のマッチドフィルタを用いてスペクトラム拡散変調信号について同期を獲得し、獲得した同期タイミングに基づいて通信制御を行う移動体通信端末装置。

要 約 書

- 本発明は、マッチドフィルタを用いた相関検出方法に関する。本発明の相関検出方法では、受信データを固定し、拡散コードをスキャンするタイプの逆拡散演算を行う。そして、拡散コードの位相シフト量に対応して一義的に定まるシンボルの区切りよりも前に位置する前半のシンボル部分と、後に位置する後半のシンボル部分の各部分について逆拡散結果を積分し、前半のシンボル部分についての積分結果を、先に実施された同様の処理の結果として保持されている、同じシンボルについての、拡散コードの同じ位相シフト量に対応する積分結果と加算し、一方、後半のシンボル部分についての積分結果は一時的に保持しておき、その保持されている積分結果を、次に同様の処理を実施した結果として得られる、同じシンボルについての、前記拡散コードの同じ位相シフト量に対応する積分結果に加算し、これによってシンボル単位の相関を検出する。

THIS PAGE BLANK

THIS PAGE BLANK (USPTO)

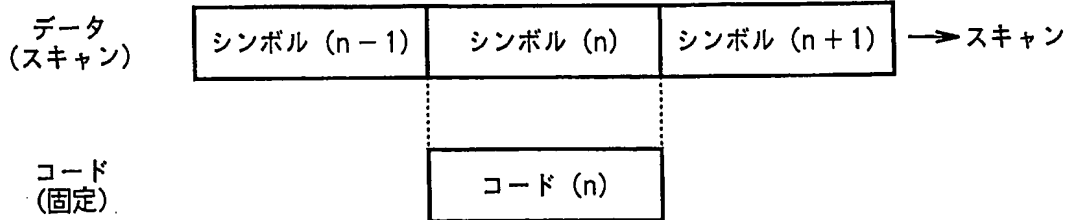


図 1A

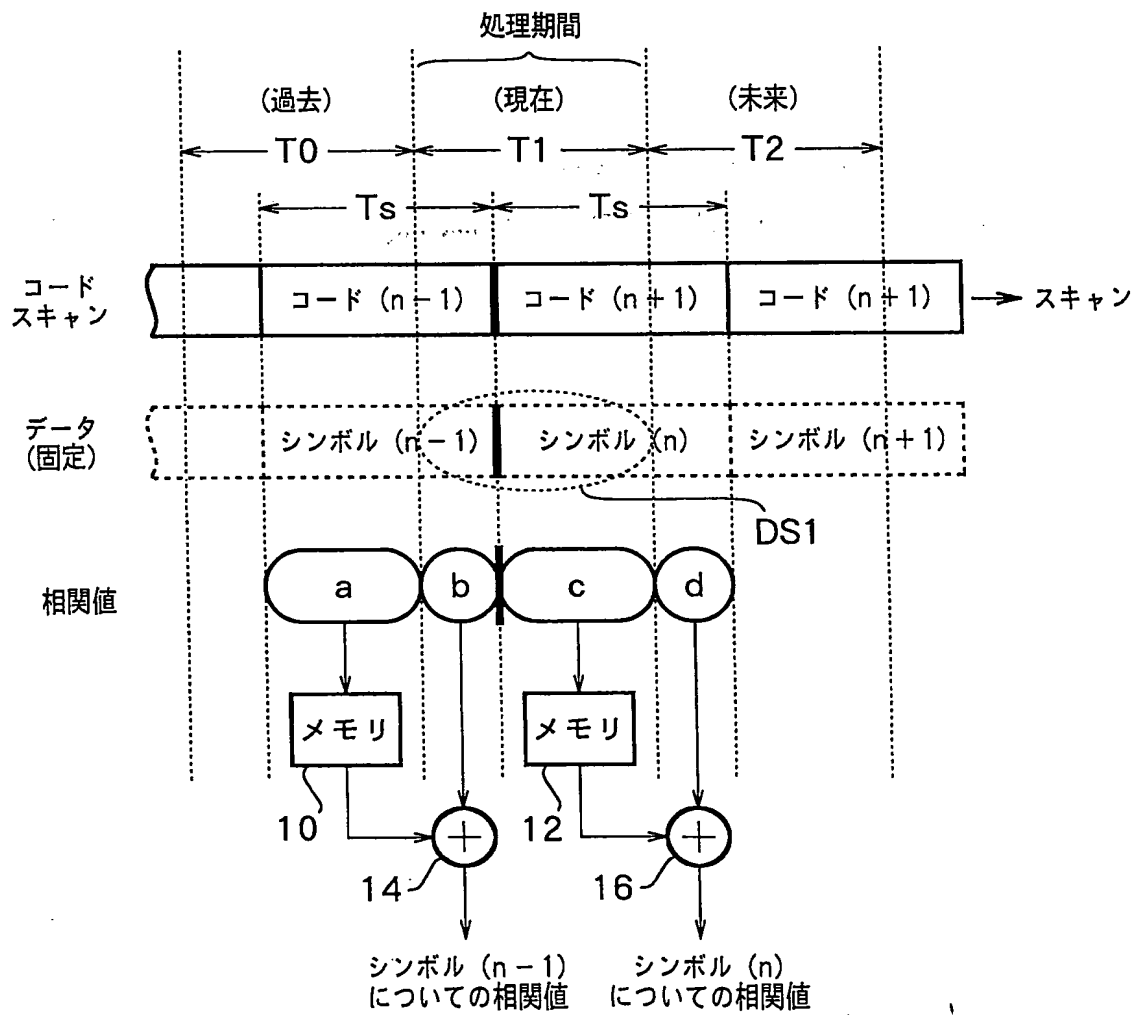


図 1B

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

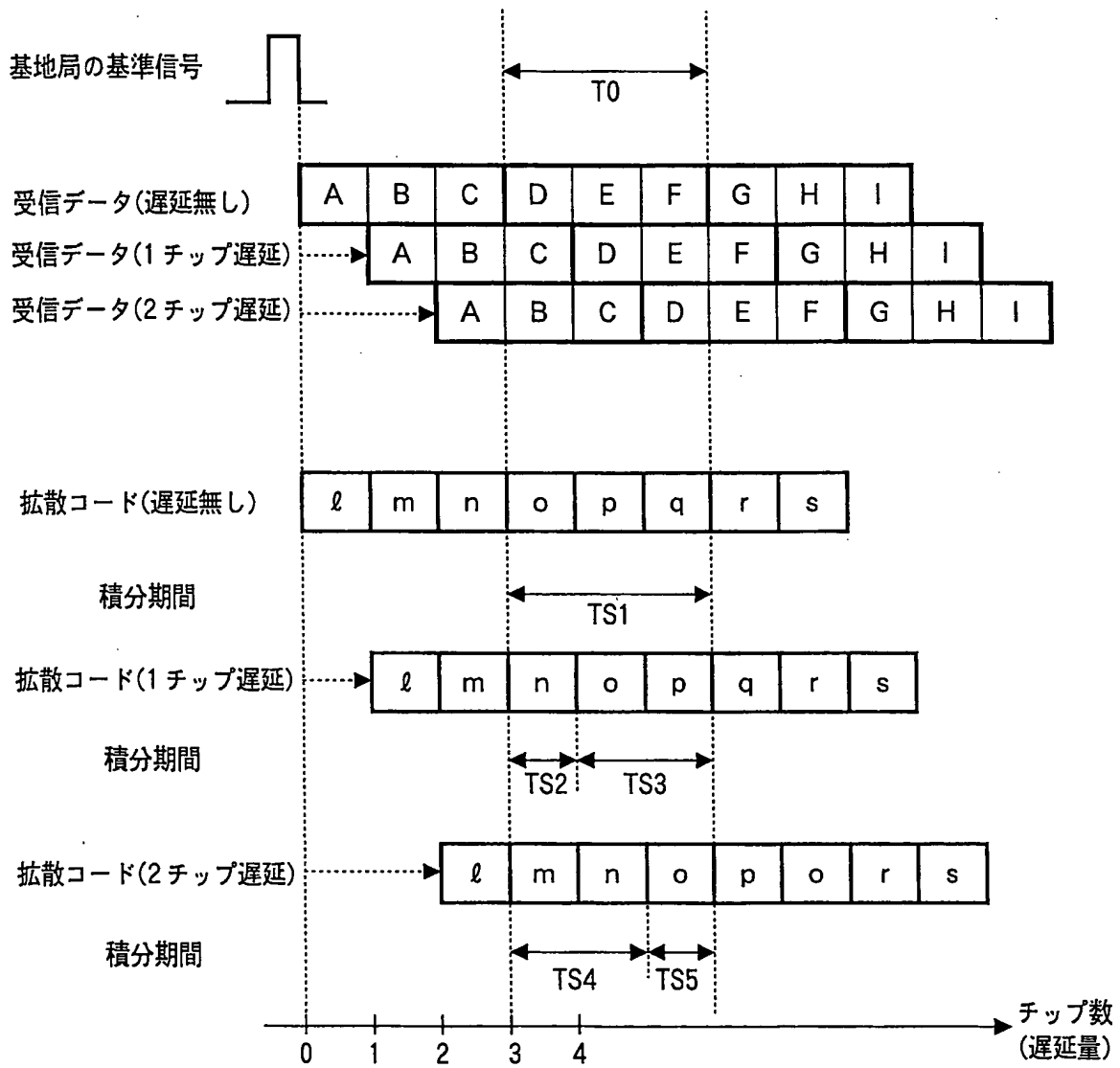


図 2

THIS PAGE BLANK (USPTO)

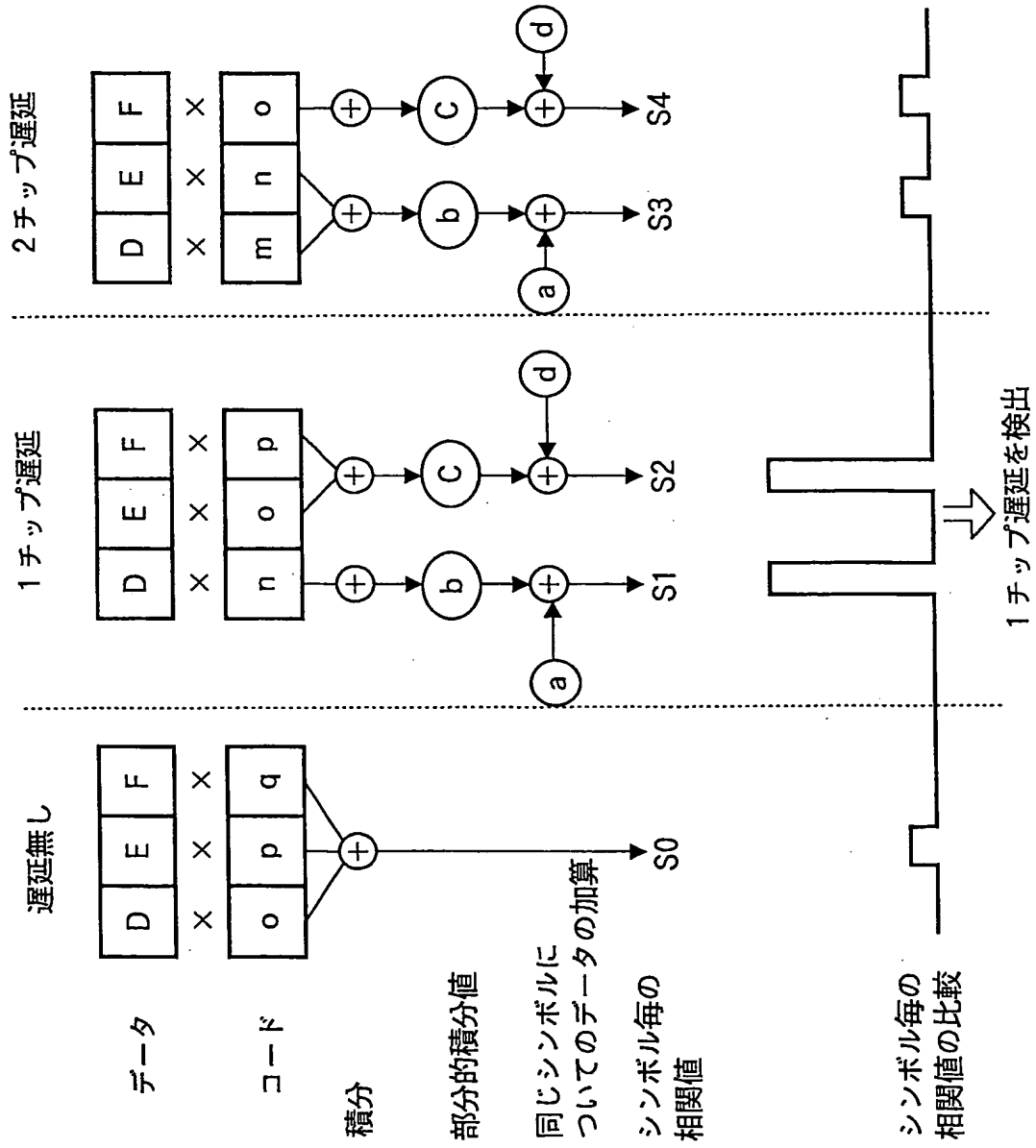


図 3

THIS PAGE BLANK (USPTO)

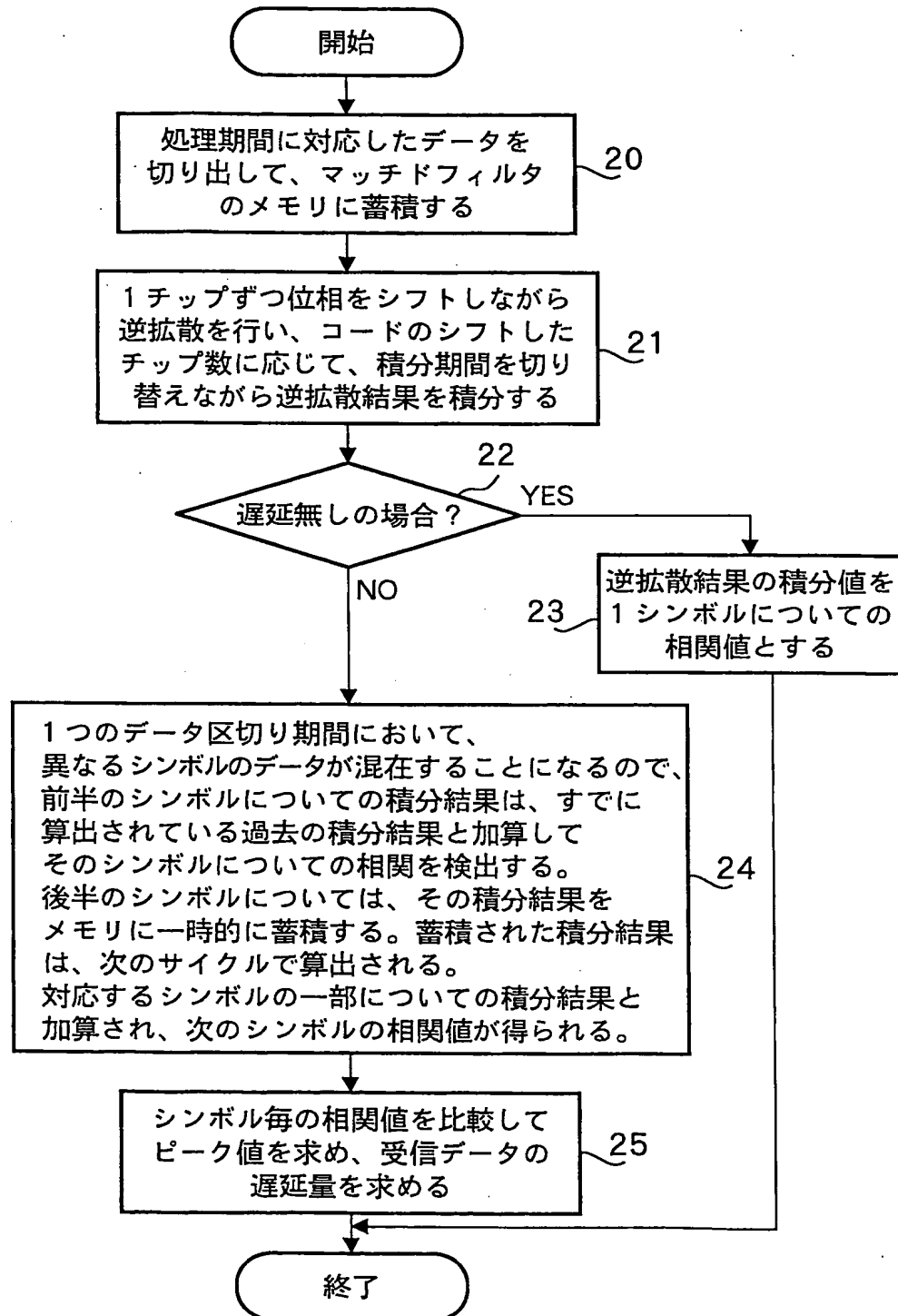


図 4

THIS PAGE BLANK (USPTO)

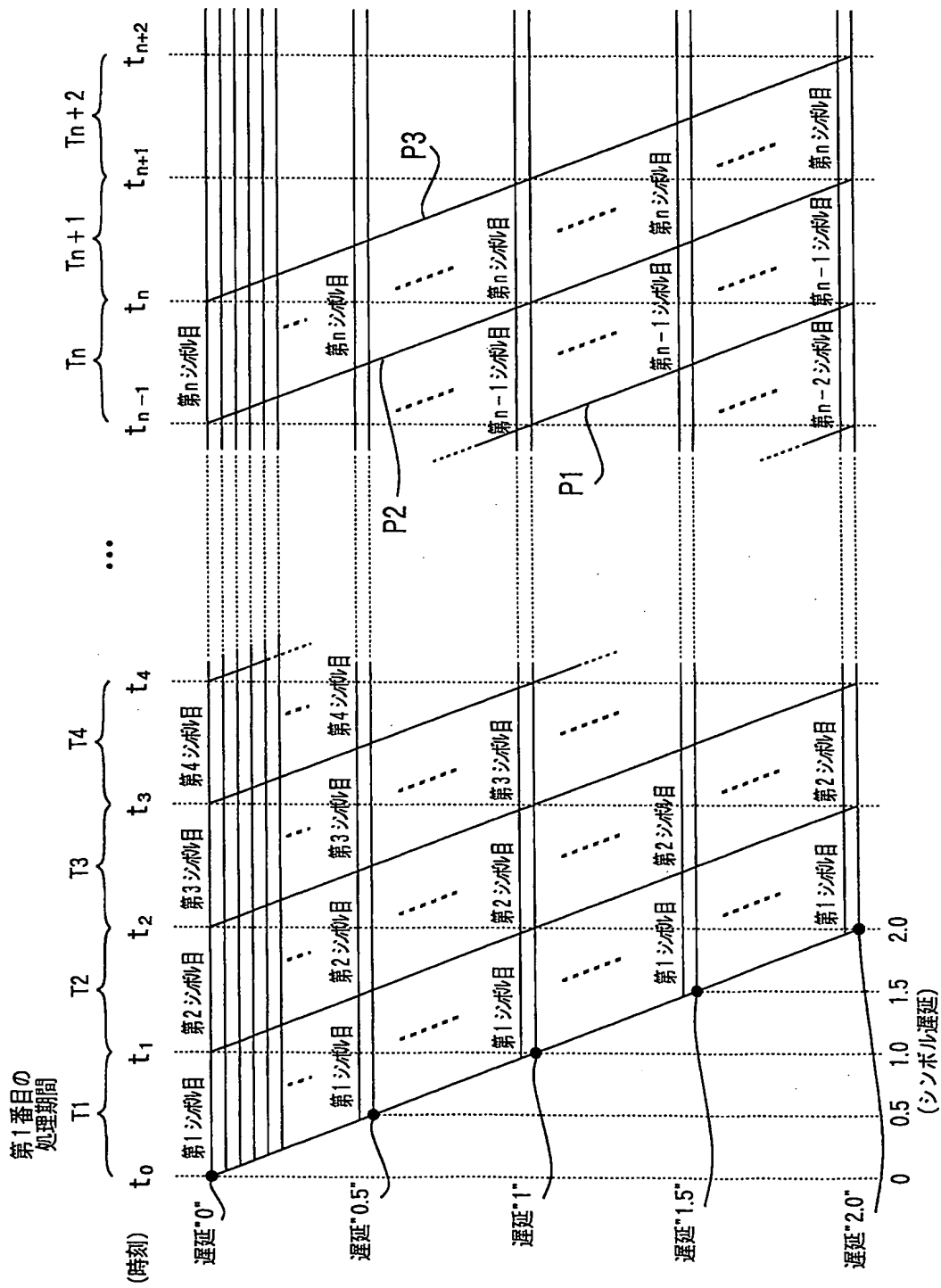
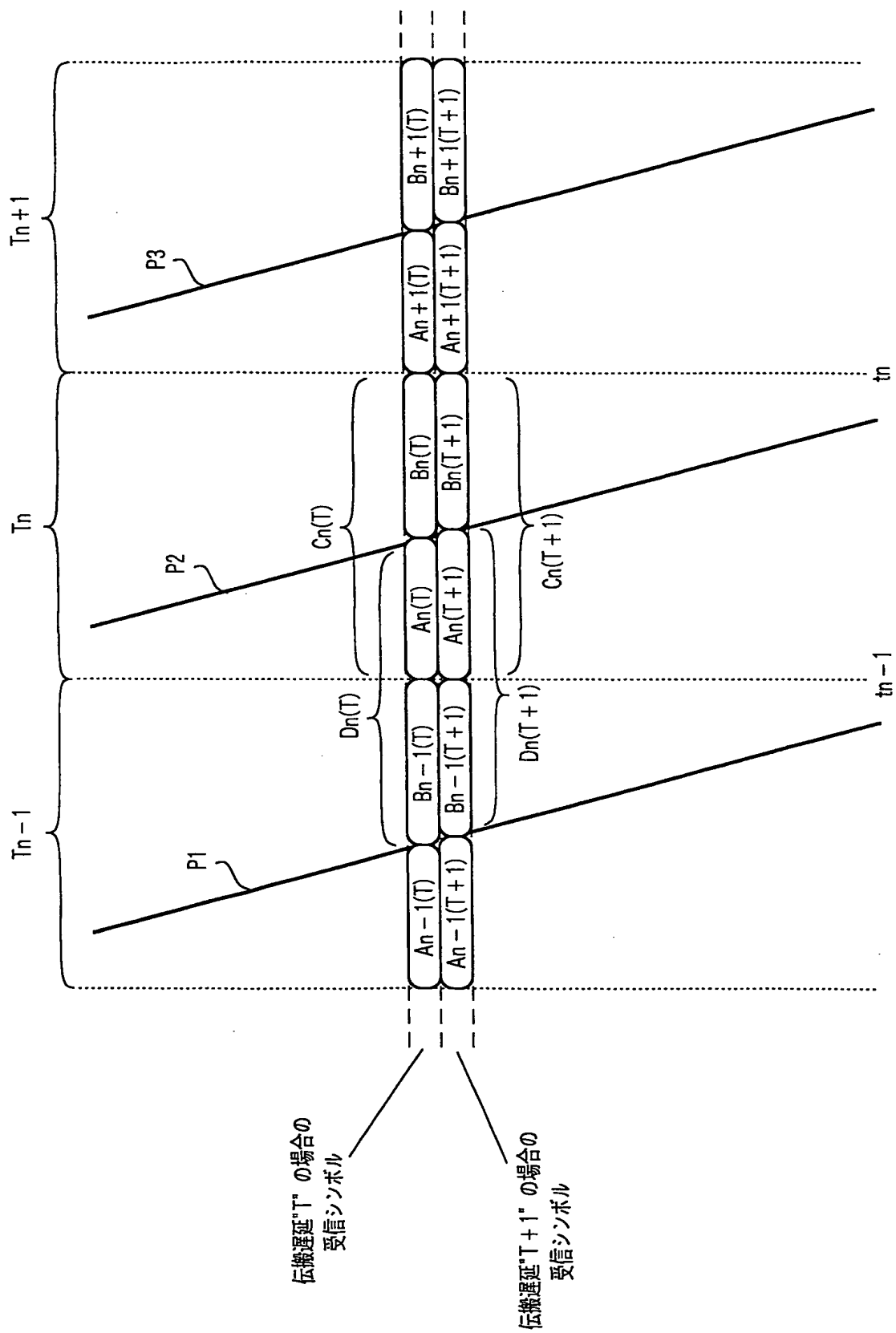
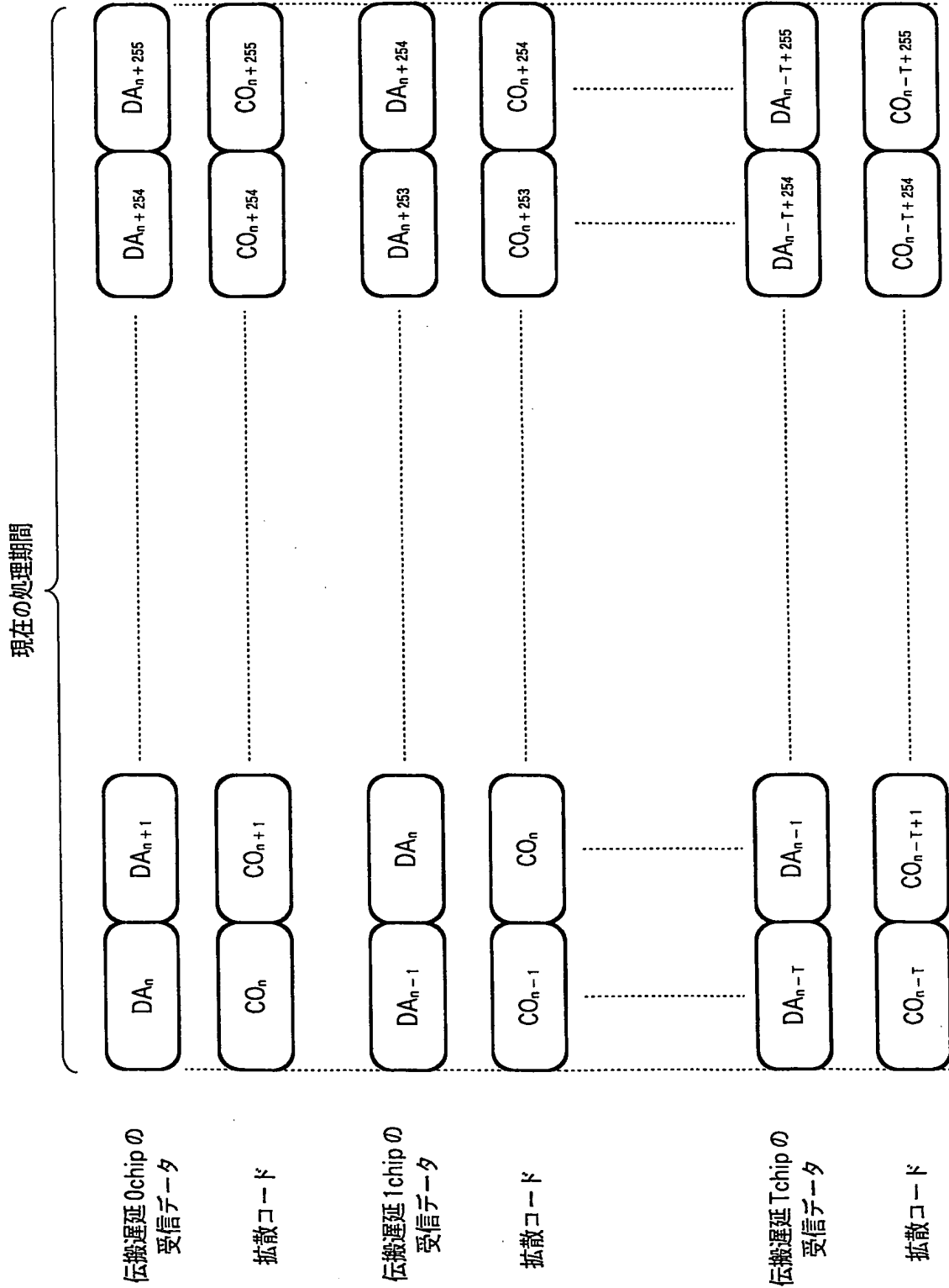


図 5

THIS PAGE BLANK (USPTO)



THIS PAGE BLANK (USPTO)



THIS PAGE BLANK (USPTO)

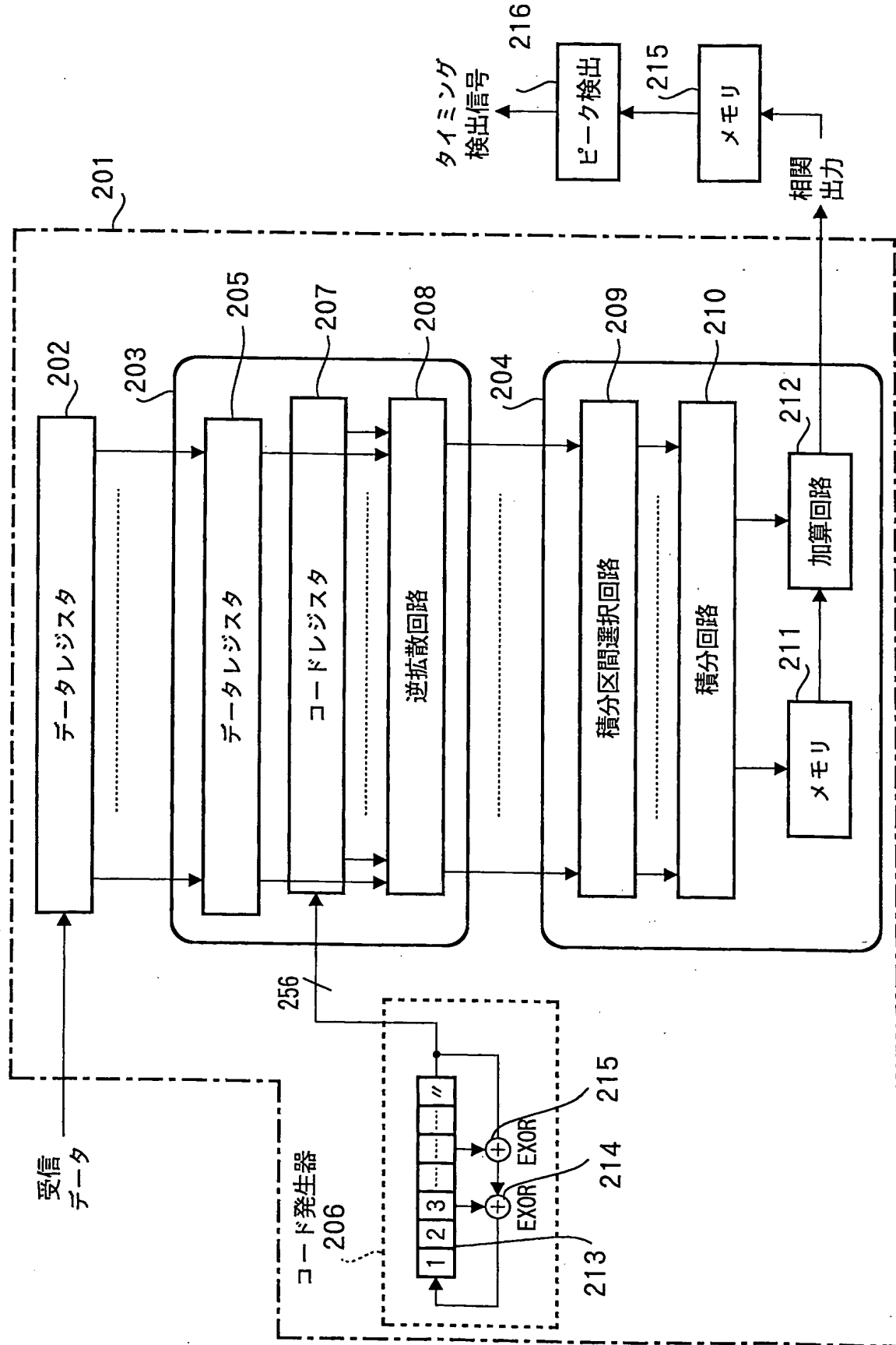


図 8

THIS PAGE BLANK (USPTO)

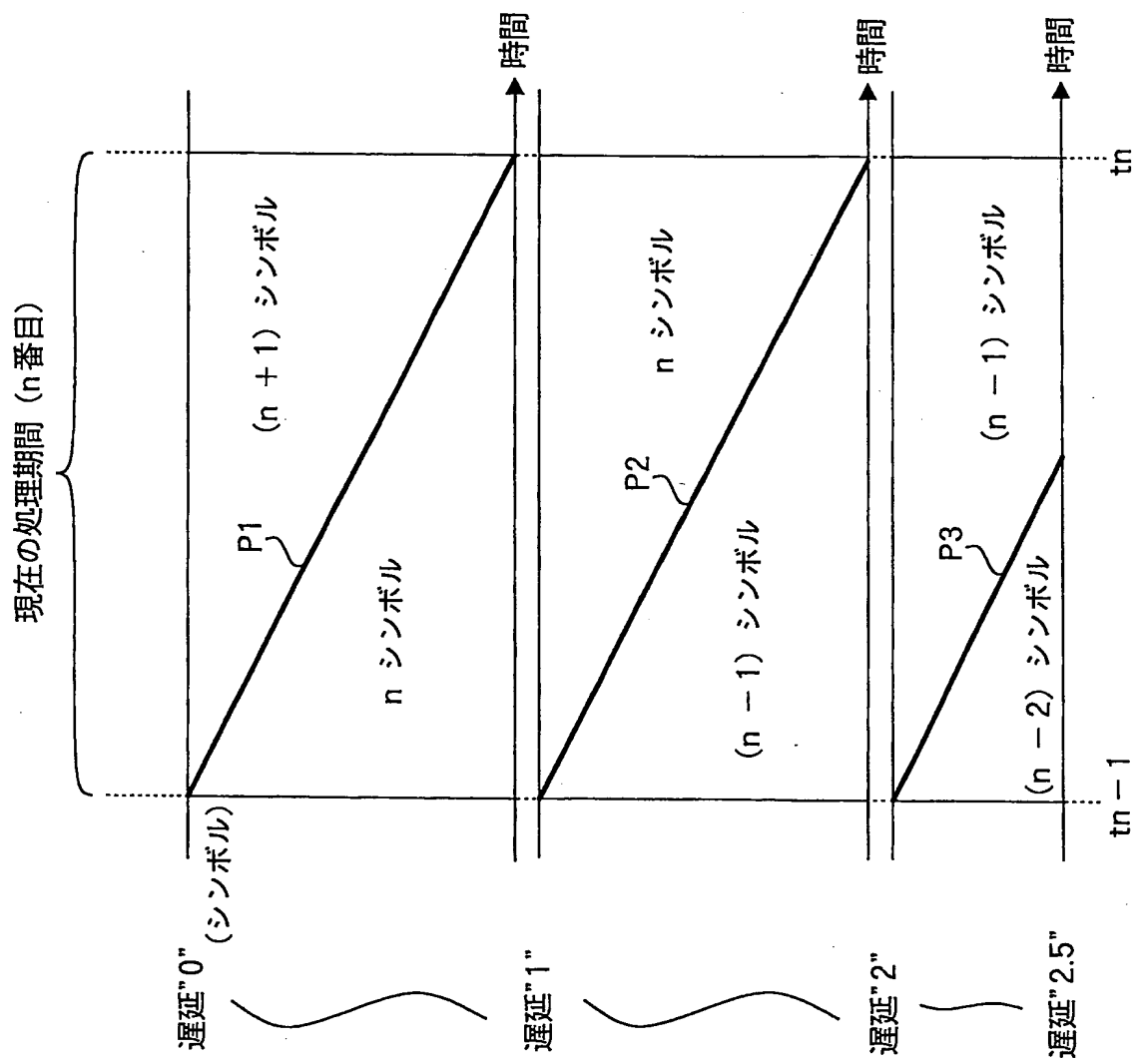


図 9

THIS PAGE BLANK (USPTO)

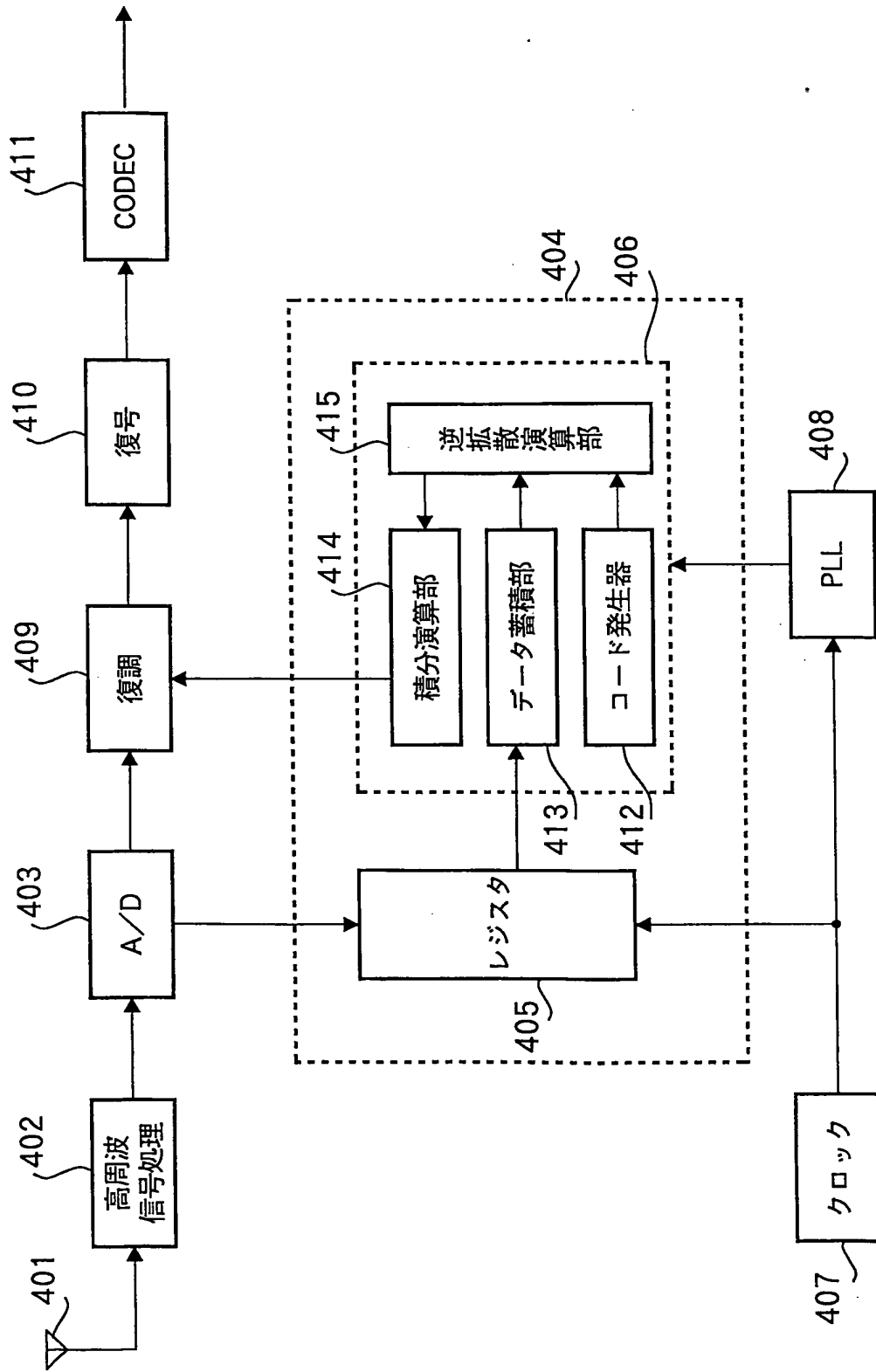


図 10

THIS PAGE BLANK (USPTO)

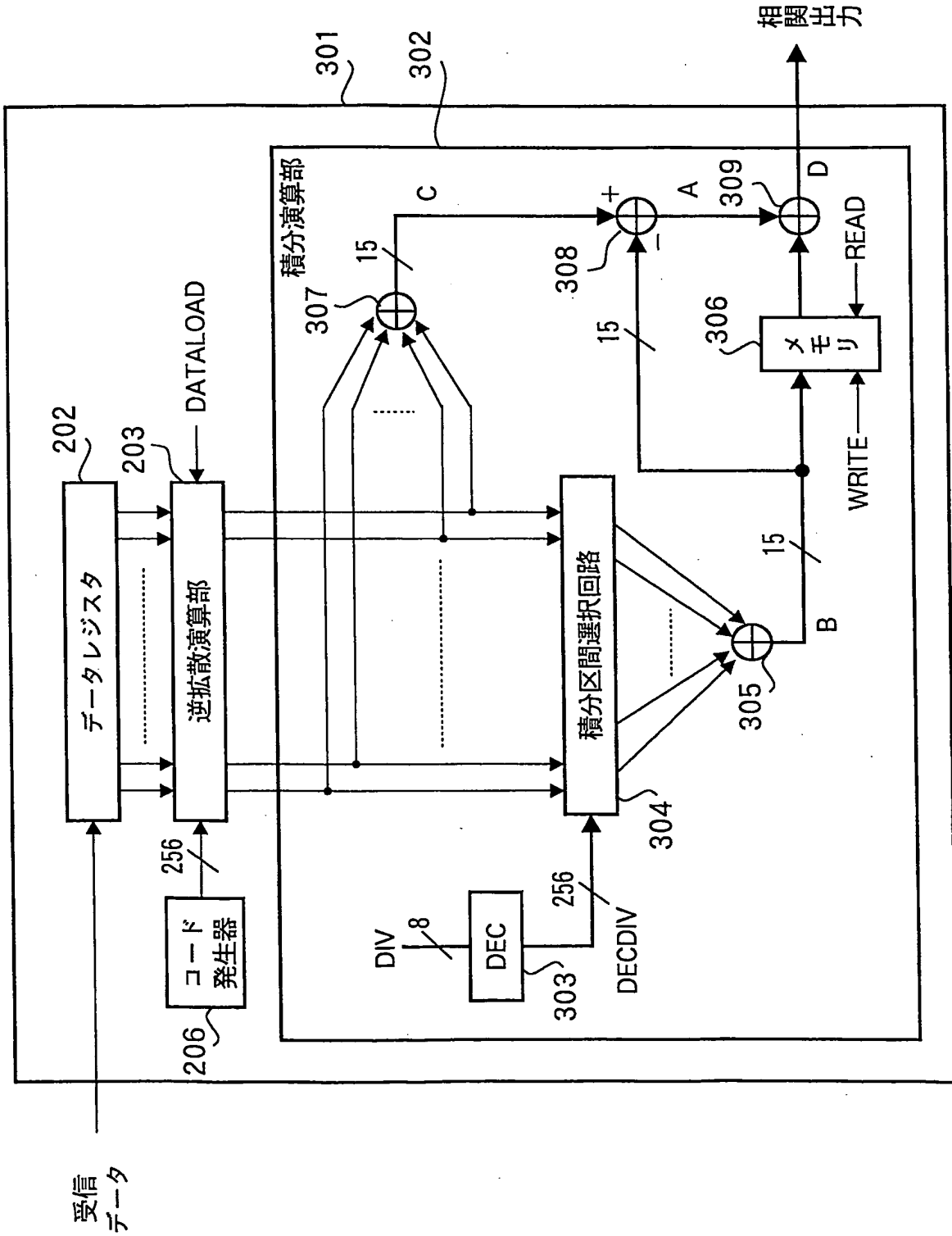


図 11

THIS PAGE BLANK (USPTO)

図 12

受信データ入力

DATALOAD

DIV

DECDIV1

DECDIV3

DECDIV3

DECDIV3

DECDIV5

DECDIV250

DECDIV251

DECDIV252

DECDIV253

DECDIV254

DECDIV255

C

B

A

ADDRESS

WRITE,READ

D

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

THIS PAGE BLANK (USPTO)

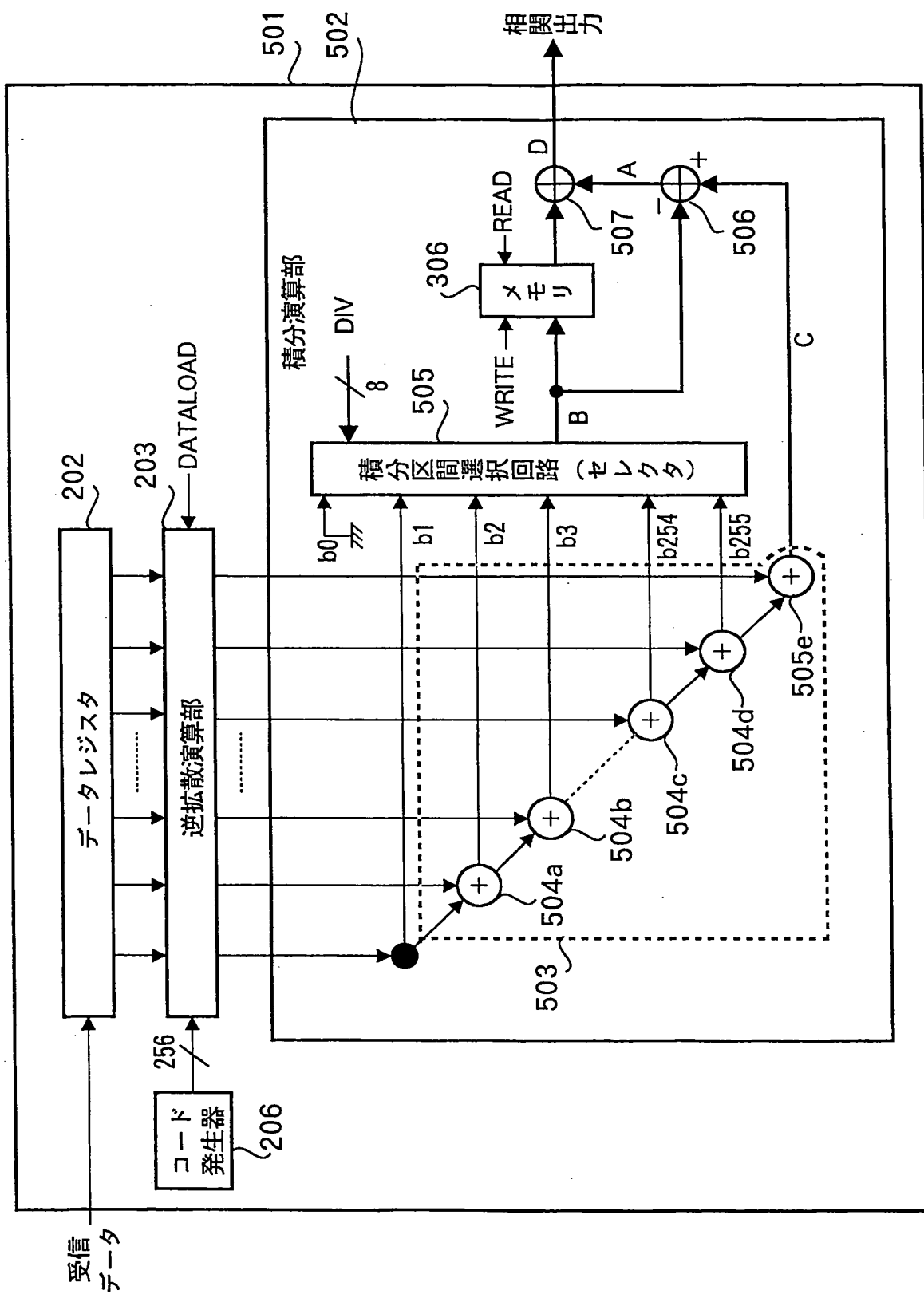


図 13

THIS PAGE BLANK (USPTO)

WRITE, READ	D															
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
ADDRESS	A															
	A2 (250)	A2 (251)	A2 (252)	A2 (253)	A2 (254)	A2 (255)	A2 (256)	A2 (257)	A2 (258)	A2 (259)	A2 (260)	A2 (261)	A2 (262)	A3 (250)	A3 (251)	A3 (252)
	A2 (250)	A2 (251)	A2 (252)	A2 (253)	A2 (254)	A2 (255)	A2 (256)	A2 (257)	A2 (258)	A2 (259)	A2 (260)	A2 (261)	A2 (262)	A3 (250)	A3 (251)	A3 (252)
B	B															
	B2 (250)	B2 (251)	B2 (252)	B2 (253)	B2 (254)	B2 (255)	B2 (256)	B2 (257)	B2 (258)	B2 (259)	B2 (260)	B2 (261)	B2 (262)	B3 (250)	B3 (251)	B3 (252)
	B2 (250)	B2 (251)	B2 (252)	B2 (253)	B2 (254)	B2 (255)	B2 (256)	B2 (257)	B2 (258)	B2 (259)	B2 (260)	B2 (261)	B2 (262)	B3 (250)	B3 (251)	B3 (252)
C	C															
	C2 (250)	C2 (251)	C2 (252)	C2 (253)	C2 (254)	C2 (255)	C2 (256)	C2 (257)	C2 (258)	C2 (259)	C2 (260)	C2 (261)	C2 (262)	C3 (250)	C3 (251)	C3 (252)
	C2 (250)	C2 (251)	C2 (252)	C2 (253)	C2 (254)	C2 (255)	C2 (256)	C2 (257)	C2 (258)	C2 (259)	C2 (260)	C2 (261)	C2 (262)	C3 (250)	C3 (251)	C3 (252)
DIV	D															
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
DATA LOAD	D															
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
受信データ入力	D															
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
	D2 (250)	D2 (251)	D2 (252)	D2 (253)	D2 (254)	D2 (255)	D2 (256)	D2 (257)	D2 (258)	D2 (259)	D2 (260)	D2 (261)	D2 (262)	D3 (250)	D3 (251)	D3 (252)
T1 T2	T															
	T2 (0)	T2 (1)	T2 (2)	T2 (3)	T2 (4)	T2 (5)	T2 (6)	T2 (7)	T2 (8)	T2 (9)	T2 (10)	T2 (11)	T2 (12)	T3 (0)	T3 (1)	T3 (2)
	T2 (0)	T2 (1)	T2 (2)	T2 (3)	T2 (4)	T2 (5)	T2 (6)	T2 (7)	T2 (8)	T2 (9)	T2 (10)	T2 (11)	T2 (12)	T3 (0)	T3 (1)	T3 (2)
T4 T5	T															
	T5 (0)	T5 (1)	T5 (2)	T5 (3)	T5 (4)	T5 (5)	T5 (6)	T5 (7)	T5 (8)	T5 (9)	T5 (10)	T5 (11)	T5 (12)	T4 (0)	T4 (1)	T4 (2)
	T5 (0)	T5 (1)	T5 (2)	T5 (3)	T5 (4)	T5 (5)	T5 (6)	T5 (7)	T5 (8)	T5 (9)	T5 (10)	T5 (11)	T5 (12)	T4 (0)	T4 (1)	T4 (2)

THIS PAGE BLANK (USPTO)

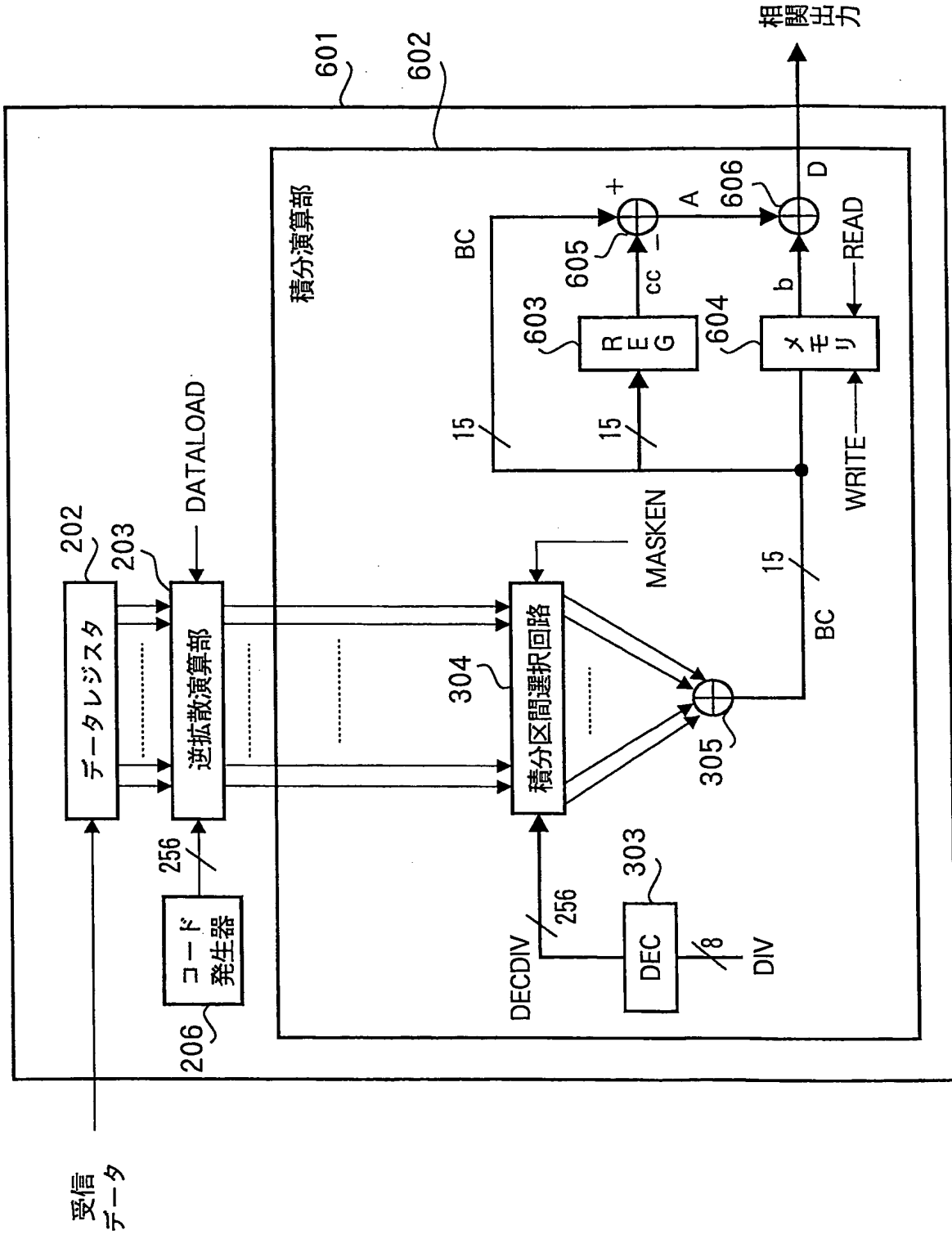


図 15

THIS PAGE BLANK (USPTO)

受信データ入力

DATALOAD

DIV

DEC DIV1

DECDIV2

DEC DIV3

DECDIV253

DEC DIV 254

DECDIV255

MASKEN

BC

ADDRESS

READ

WRITE

9

8

A

$$T_1 \quad T_2 \quad T_3 \quad T_4$$

T5 T6

D3(258)	A3(258)	C3(258)	B2(258)	258	B3(258)	C3(258)	2
D3(257)	A3(257)	C3(257)	B2(257)	257	B3(257)	C3(257)	1
D3(256)	A3(256)	C3(256)	B2(256)	256	B3(256)	C3(256)	0
D3(255)	A3(255)	C3(255)	B2(255)	255	B3(255)	C3(255)	255
D3(254)	A3(254)	C3(254)	B2(254)	254	B3(254)	C3(254)	254
D3(253)	A3(253)	C3(253)	B2(253)	253	B3(253)	C3(253)	253
D3(2)	A3(2)	C3(2)	B2(2)	2	B3(2)	C3(2)	2
D3(1)	A3(1)	C3(1)	B2(1)	1	B3(1)	C3(1)	1
D3(0)	A3(0)	C3(0)	B2(0)	0	B3(0)	C3(0)	0
D2(511)	A2(511)	C2(511)	B1(511)	511	B2(511)	C2(511)	255
D2(510)	A2(510)	C2(510)	B1(510)	510	B2(510)	C2(510)	254
D2(509)	A2(509)	C2(509)	B1(509)	509	B2(509)	C2(509)	253
D2(258)	A2(258)	C2(258)	B1(258)	258	B2(258)	C2(258)	2
D2(257)	A2(257)	C2(257)	B1(257)	257	B2(257)	C2(257)	1
D2(256)	A2(256)	C2(256)	B1(256)	256	B2(256)	C2(256)	0
D2(255)	A2(255)	C2(255)	B1(255)	255	B2(255)	C2(255)	255
D2(254)	A2(254)	C2(254)	B1(254)	254	B2(254)	C2(254)	254
D2(253)	A2(253)	C2(253)	B1(253)	253	B2(253)	C2(253)	253
D2(2)	A2(2)	C2(2)	B1(2)	2	B2(2)	C2(2)	2
D2(1)	A2(1)	C2(1)	B1(1)	1	B2(1)	C2(1)	1
D2(0)	A2(0)	C2(0)	B1(0)	0	B2(0)	C2(0)	0

THIS PAGE BLANK (USPTO)

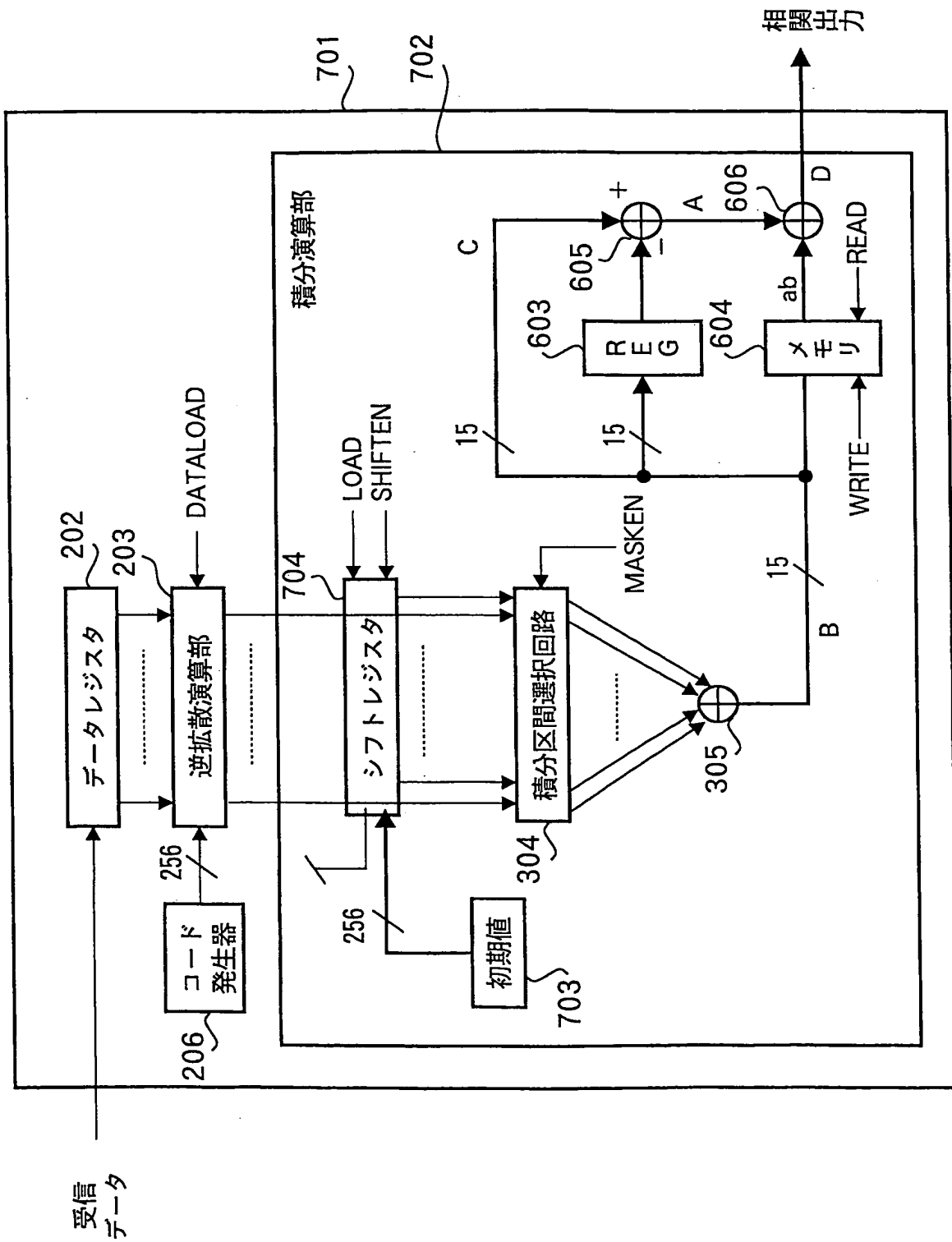
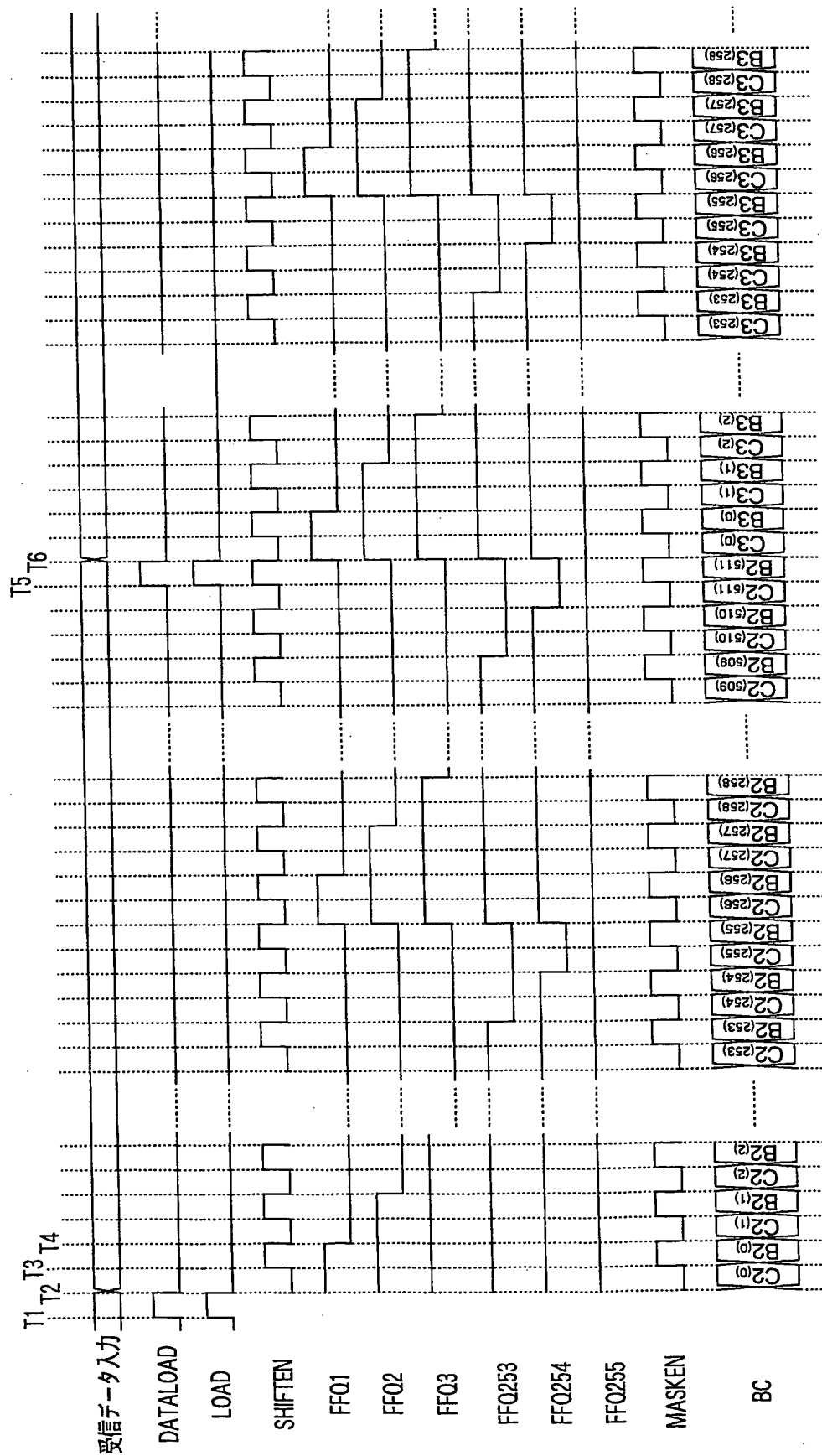


図 17

THIS PAGE BLANK (USPTO)



THIS PAGE BLANK (USPTO)

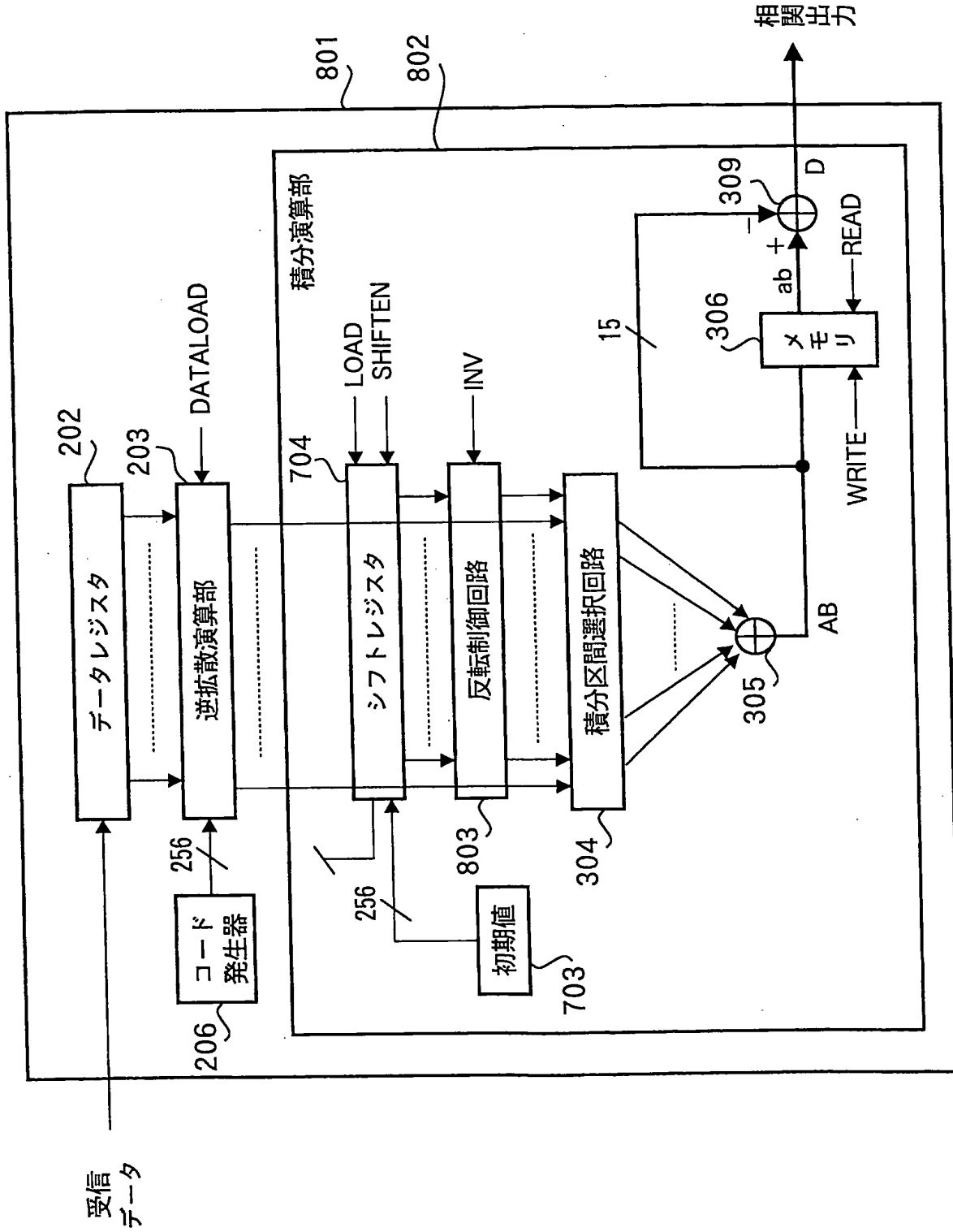
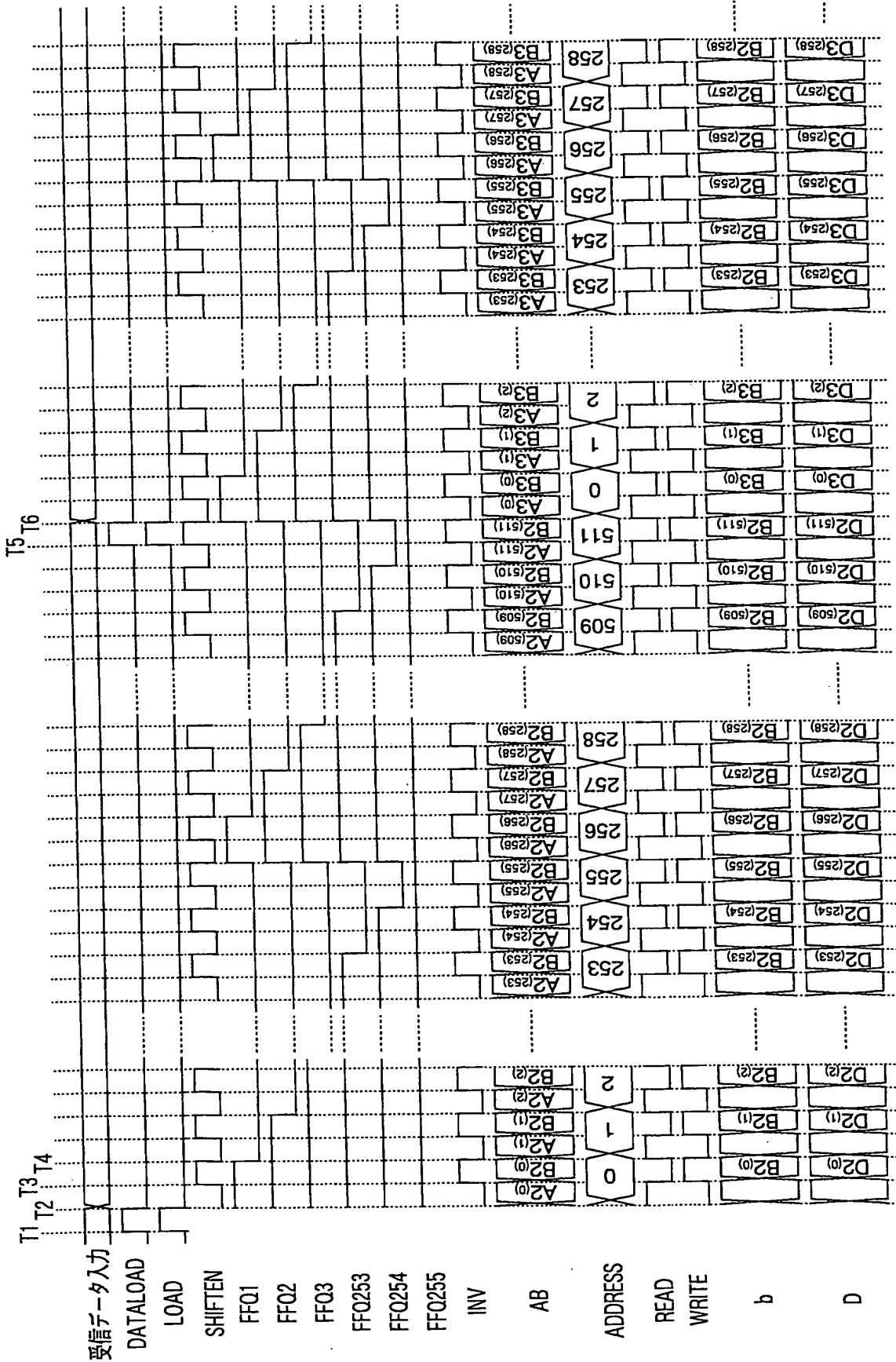


図 19

THIS PAGE BLANK (USPTO)

20



THIS PAGE BLANK (USPTO)

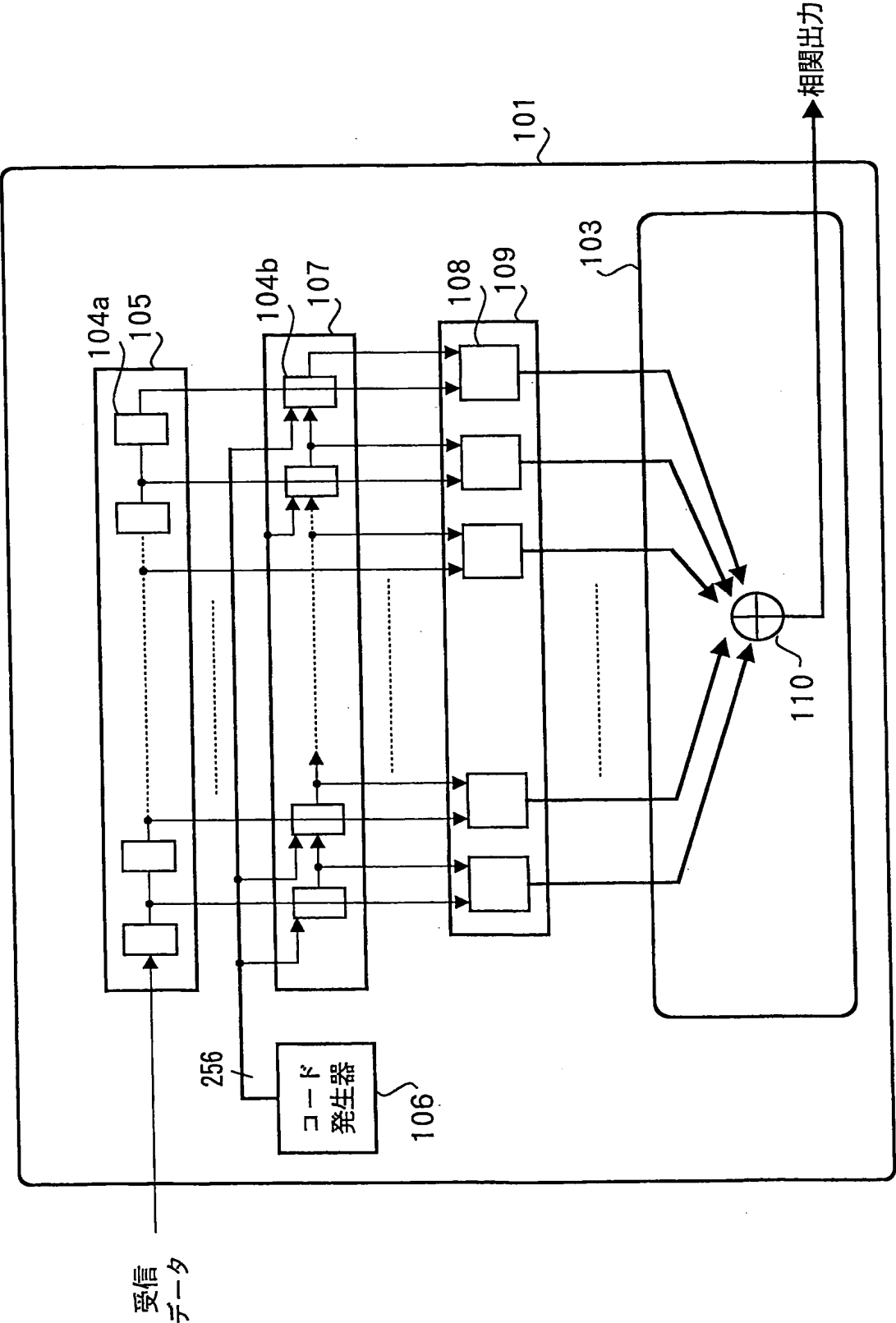


図 21

THIS PAGE BLANK (USPTO)

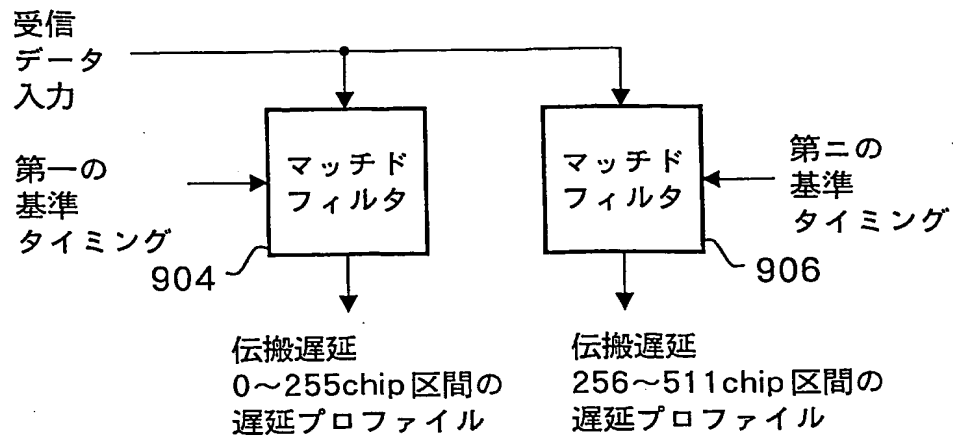


図 22A

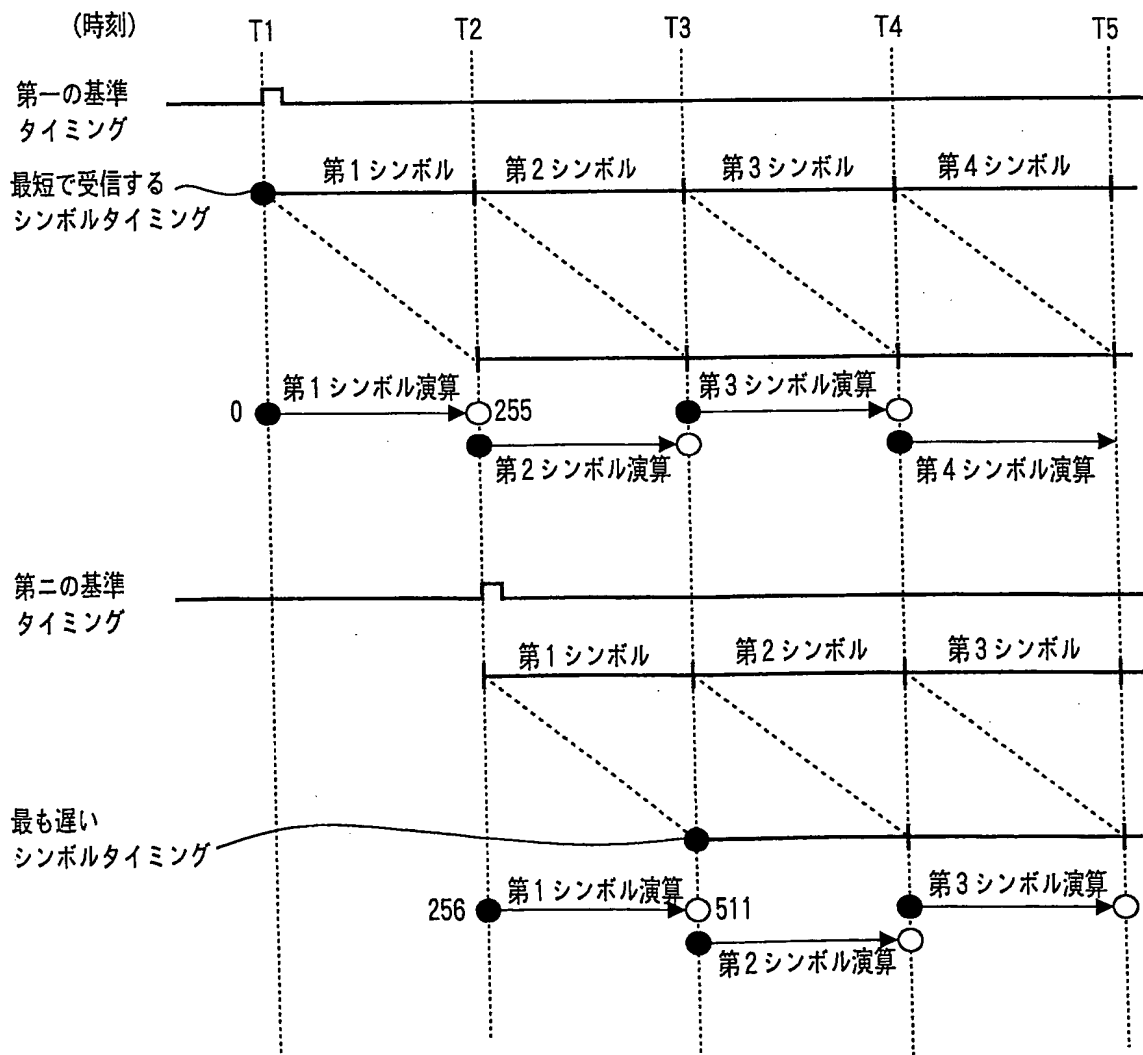


図 22B

THIS PAGE BLANK (USPTO)